NS7520 Hardware Reference

Making DEVICE NETWORKING easy

NS7520 Hardware Reference

Part number/version: 90000353_D
Release date: March 2006
www.digi.com

Digi International
11001 Bren Road East
Minnetonka, MN 55343 U.S.A.
United States: +1 877 912-3444
Other locations: +1 952 912-3444

www.digi.com/support/
www.digi.com

Contents

Using This Guide

Review this section for basic information about the guide you are using, as well as general support and contact information.

About this guide

This guide provides information about the NS7520 32-bit networked microprocessor. The NS7520 is part of the NET+ARM line of SoC (System-on-Chip) products, and supports high-bandwidth applications for intelligent networked devices.

The NET+ARM family is part of the NET+Works integrated product family, which includes the NET+OS network software suite.

Who should read this guide

This guide is for hardware developers, system software developers, and applications programmers who want to use the NS7520 for development.

To complete the tasks described in this guide, you must:

- Understand the basics of hardware and software design, operating systems, and microprocessor design.
- Understand the NS7520 architecture.

What's in this guide

This table shows where you can find specific information in this guide:

To read about	See
NS7520 key features	Chapter 1, "About the NS7520"
NS7520 ball grid array assignments & packaging	Chapter 2, "Pinout and Packaging"
NS7520 CPU and ARM Thumb concept	Chapter 3, "Working with the CPU"
BBus functionality	Chapter 4, "BBus Module"
System functionality	Chapter 5, "SYS Module"
General (GEN) module functionality	Chapter 6, "GEN Module"
How the NS7520 can be configured to interface with different types of memory devices	Chapter 7, "Memory Controller Module"
DMA controller, supported DMA channels, and internal and external DMA transfers	Chapter 8, "DMA Module"
Ethernet controller module	Chapter 9, "Ethernet Module"
Serial channels A and B	Chapter 10, "Serial Controller Module"
NS7520 timing information and diagrams	Chapter 11, "Electrical Characteristics"

Conventions used in this guide

This table describes the typographic conventions used in this guide:

This convention	Is used for
italic type	Emphasis, new terms, variables, and document titles.
`monospaced type`	Filenames, pathnames, and code examples.
_ (underscore)	Defines a signal as being active low.
'b	Indicates that the number following this indicator is in binary radix
'd	Indicates that the number following this indicator is in decimal radix
'h	Indicates that the number following this indicator is in hexadecimal radix

Related documentation

NS7520 Jumpers and Components provides a hardware description of the NET+Works Development Board, and includes information about jumpers, connectors, switches, and interface configurations, as well as development board diagrams.

Review the documentation CD-ROM that came with your development kit for information on third-party products and other components.

See the NET+OS software documentation for information appropriate to the chip you are using.

Documentation updates

Digi occasionally provides documentation updates on the Web site (www.digi.com/support).

Be aware that if you see differences between the documentation you received in your package and the documentation on the Web site, the Web site content is the latest version.

Customer support

To get help with a question or technical problem with this product, or to make comments and recommendations about our products or documentation, use the contact information listed in this table:

For	Contact information
Technical support	United States: +1 877 912-3444 Other locations: +1 952 912-3444 www.digi.com/support www.digi.com

About the NS7520

C H A P T E R 1

This chapter provides an overview of the NS7520. The NS7520 is a high-performance, highly integrated, 32-bit system-on-a-chip ASIC designed for use in intelligent networked devices and Internet appliances. The NS7520 is based on the standard architecture in the NET+ARM family of devices.

NET+ARM is the hardware foundation of the NET+Works family of integrated hardware and software solutions for device networking. These comprehensive platforms include drivers, popular operating systems, networking software, development tools, APIs, and complete development boards.

NS7520 Features

The NS7520 can support most any networking scenario, and includes a 10/100 BaseT Ethernet MAC and two independent serial ports (each of which can run in UART or SPI mode).

The CPU is an ARM7TDMI (ARM7) 32-bit RISC processor core with a rich complement of support peripherals and memory controllers, including:

- Glueless connection to different types of memory; for example, flash, SDRAM, EEPROM, and others.
- Programmable timers
- 13-channel DMA controller
- External bus expansion module
- 16 general-purpose I/O (GPIO) pins

Key features and operating modes of the major NS7520 modules

- CPU core
  - ARM7 32-bit RISC processor
  - 32-bit internal bus
  - 32-bit ARM mode and 16-bit Thumb mode
  - 15 general-purpose 32-bit registers
  - 32-bit program counter (PC) and status register
  - Five supervisor modes, one user mode
- 13-channel DMA controller
  - Two channels dedicated to Ethernet transmit and receive
  - Four channels dedicated to two serial modules' transmit and receive
  - Four channels for external peripherals (only two channels — either 3 and 5 or 4 and 6 — can be configured at one time)
  - Three channels available for memory-to-memory transfers
  - Flexible buffer management

- General-purpose I/O pins
  - 16 programmable GPIO interface pins
  - Four pins programmable with level-sensitive interrupt
- Serial ports
  - Two fully independent serial ports (UART, SPI)
  - Digital phase lock loop (DPLL) for receive clock extractions
  - 32-byte transmit/receive FIFOs
  - Internal programmable bit-rate generators
  - Bit rates 75–230400 in 16X mode
  - Bit rates 1200 bps–4 Mbps in 1X mode
  - Flexible baud rate generator, external clock for synchronous operation
  - Receive-side character and buffer gap timers
  - Four receive-side data match detectors
- Power and operating voltages
  - 500 mW maximum at 55 MHz (all outputs switching)
  - 418 mW maximum at 46 MHz (all outputs switching)
  - 291 mW maximum at 36 MHz (all outputs switching)
  - 3.3 V — I/O
  - 1.5 V — Core
- Integrated 10/100 Ethernet MAC
  - 10/100 Mbps MII-based PHY interface
  - 10 Mbps ENDEC interface
  - Support for TP-PMD and fiber-PMD devices
  - Full-duplex and half-duplex modes
  - Optional 4B/5B coding
  - Station, broadcast, and multicast address detection filtering
  - 512-byte transmit FIFO, 2 Kbyte receive FIFO
  - Intelligent receive-side buffer size selection

- Programmable timers
  - Two independent timers (2μs–20.7 hours)
  - Watchdog timer (interrupt or reset on expiration)
  - Programmable bus monitor or timer
- Operating frequency
  - 36, 46, or 55 MHz internal clock operation from 18.432 MHz quartz crystal or crystal oscillator
  - f_{MAX} = 36, 46, or 55 MHz (grade-dependent)
  - System clock source by external quartz crystal or crystal oscillator, or clock signal
  - Programmable PLL, which allows a range of operating frequencies from 10 to f_{MAX}
  - Maximum operating frequency from external clock or using PLL multiplication f_{MAX}
- Bus interface
  - Five independent programmable chip selects with 256 Mb addressing per chip select
  - All chip selects support SRAM, FP/EDO DRAM, SDRAM, flash, and EEPROM without external glue
  - Supports 8-, 16-, and 32-bit peripherals
  - External address decoding and cycle termination
  - Dynamic bus sizing
  - Internal DRAM/SDRAM controller with address multiplexer and programmable refresh frequency
  - Internal refresh controller (CAS before RAS)
  - Burst-mode support
  - 0–63 wait states per chip select
  - Address pins that configure chip operating modes; see "NS7520 bootstrap initialization" on page 60.

NS7520 module block diagram

Figure 1 is an overview of the NS7520, including all the modules.

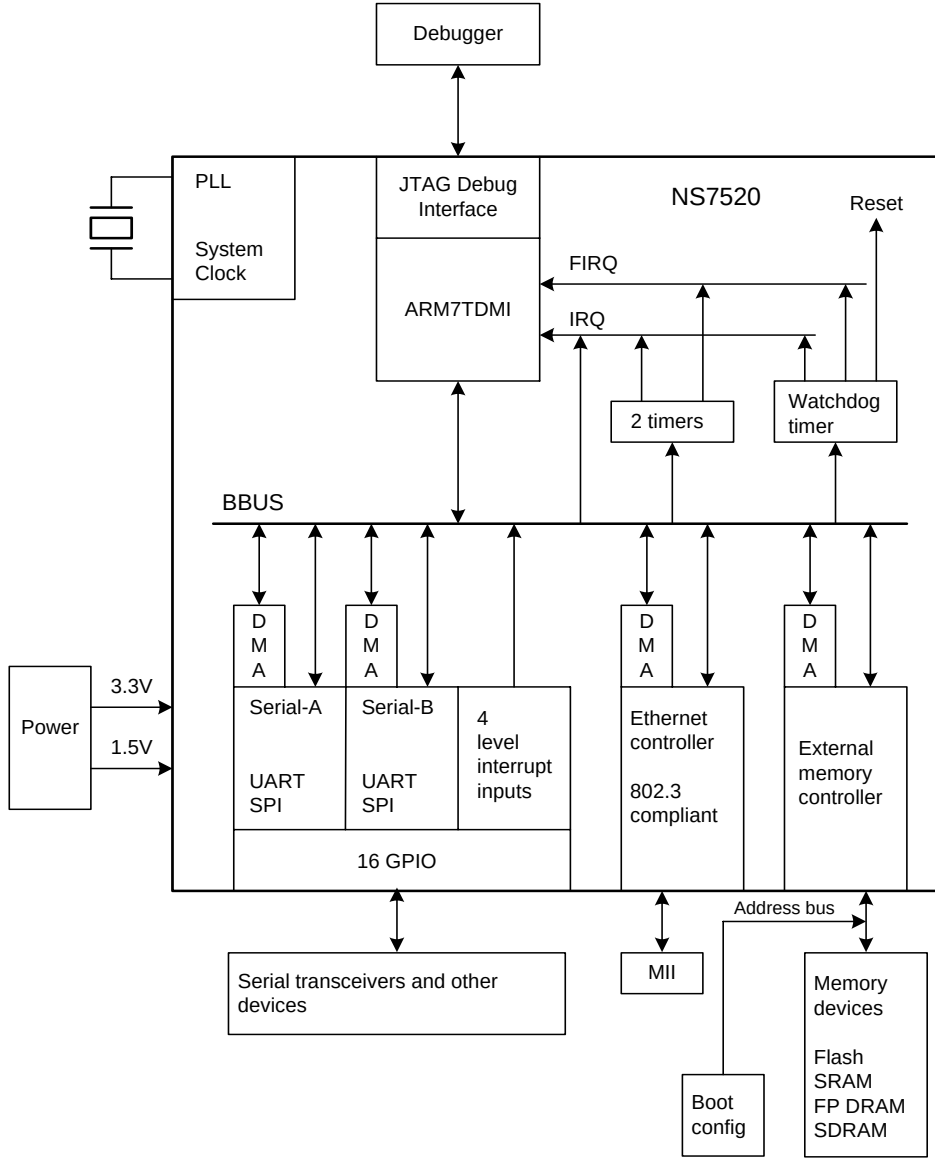


Figure 1: NS7520 overview

Operating frequency

The NS7520 is available in grades operating at three maximum operating frequencies: 36 MHz, 46 MHz, and 55 MHz. The operating frequency is set during bootstrap initialization, using pins A[8:0]. These address pins load the PLL settings register on powerup reset. A[8:7] determines IS (charge pump current); A[6:5] determines FS (output divider), and A[4:0] defines ND (PLL multiplier). Each bit in A[8:0] can be set individually.

See "Setting the PLL frequency," beginning on page 54, for more detailed information.

Pinout and Packaging

CHAPTER 2

The NS7520 can be used in any embedded environment requiring networking services in an Ethernet LAN. The NS7520 contains an integrated ARM RISC processor, 10/100 Ethernet MAC, serial ports, memory controllers, and parallel I/O. The NS7520 can interface with another processor using a register or shared RAM interface. The NS7520 provides all the tools required for any embedded networking application.

Packaging

Table 1 provides the NS7520 packaging dimensions. Figure 2 shows the pinout and NS7520 dimensions. Figure 3 shows the NS7520 BGA layout.

Symbol	Min	Nom	Max
A	—	—	1.4
A1	0.35	0.40	0.45
A2	—	—	0.95
b	0.45	0.50	0.55
D		13.0 BSC	
D1		11.2 BSC	
E		13.0 BSC	
E1		11.2 BSC	
e		0.8 BSC	
aaa		0.1	

Table 1: NS7520 packaging dimensions

177 PFBGA

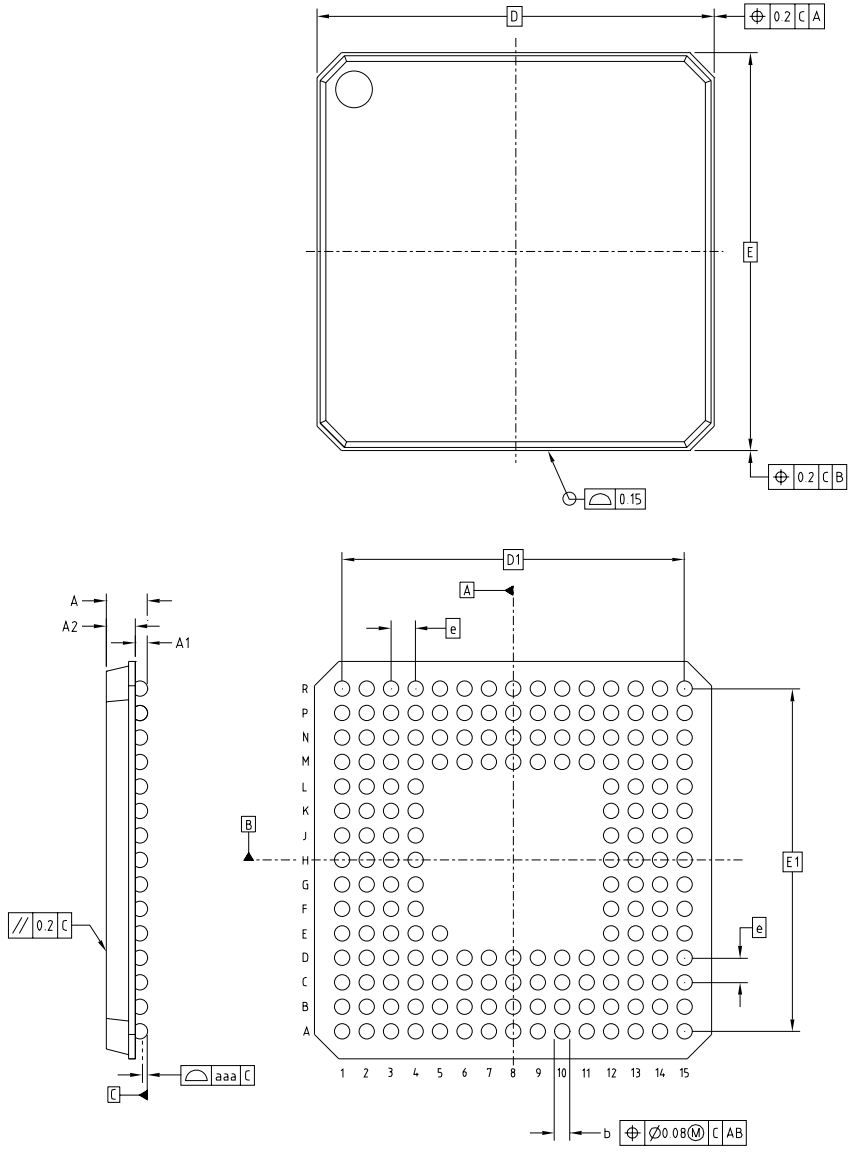


Figure 2: NS7520 pinout and dimensions

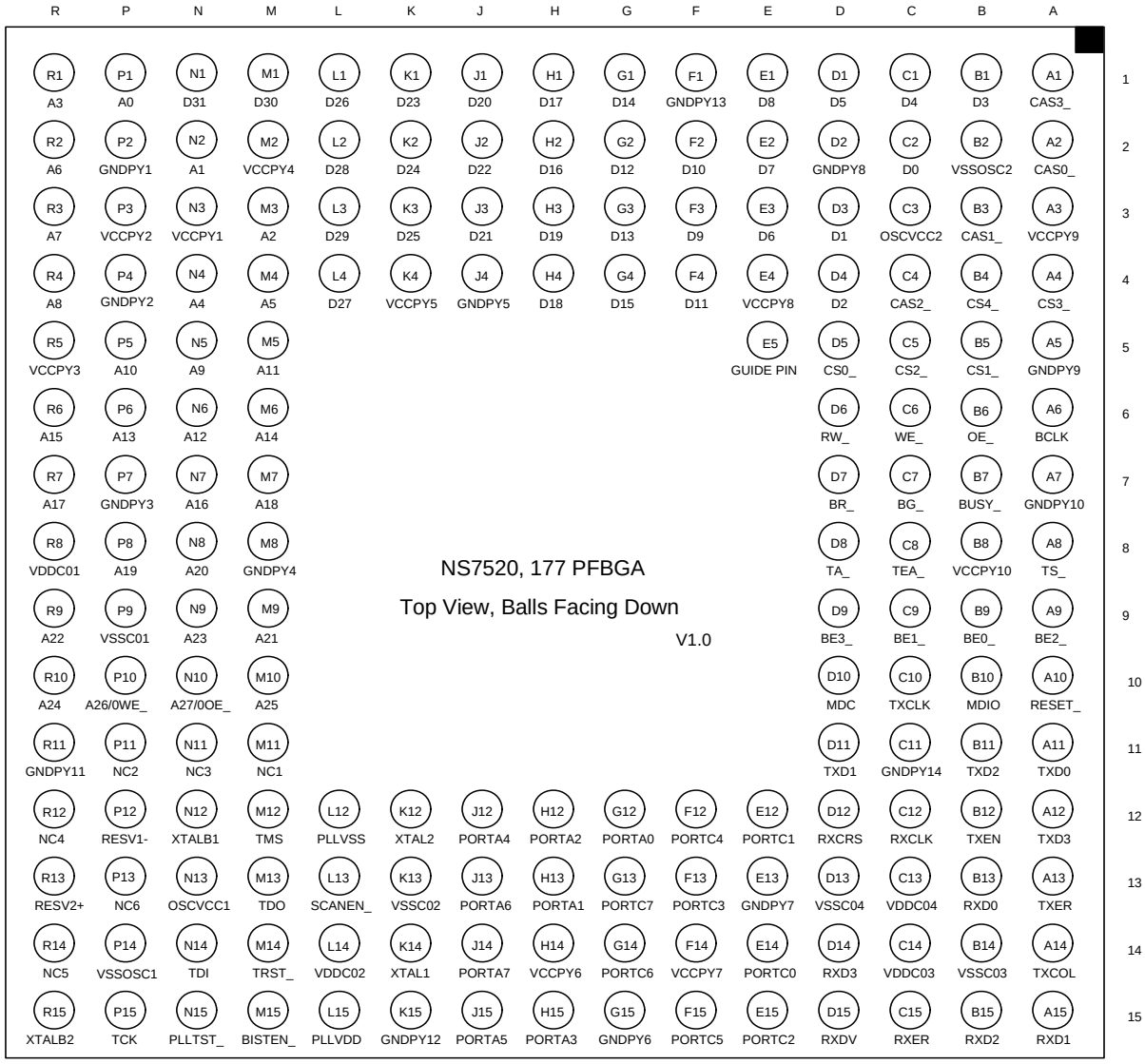


Figure 3: NS7520 BGA layout

Pinout detail tables and signal descriptions

Each pinout table applies to a specific interface and contains the following information:

Column	Description
Signal	The pin name for each I/O signal. Some signals have multiple function modes and are identified accordingly. The mode is configured through firmware using one or more configuration registers.
Pin	The pin number assignment for a specific I/O signal. ■ **U** next to the pin number indicates that the pin is a pullup resistor (input current source). ■ **D** next to the pin number indicates that the pin is a pulldown resistor (input current sink). ■ No value next to the pin indicates that the pin has neither a pullup nor pulldown resistor. See Figure 28, "Internal pullup characteristics," on page 264 and Figure 29, "Internal pulldown characteristics," on page 264 for an illustration of the characteristics of these pins. Use the figures to select the appropriate value of the complimentary resistor to drive the signal to the opposite logic state. For those pins with no pullup or pulldown resistor, you must select the appropriate value per your design requirements.
_	An underscore (bar) indicates that the pin is *active low*.
I/O	The type of signal — input, output, or input/output.
OD	The output drive strength of an output buffer. The NS7520 uses one of three drivers: ■ 2 mA ■ 4 mA ■ 8 mA

Notes:

- NO CONNECT as a description for a pin means *do not connect to this pin*.
- The 177th pin (package ball) is for alignment of the package on the PCB.

System bus interface

Symbol		Pin	I/O	OD	Description	
BCLK		A6	O	8	Synchronous bus clock	
External bus	Other				External bus	Other
ADDR27	CS0OE_	N10 U	I/O	4	Addr bit 27	Logical AND of CS0_ and OE_
ADDR26	CS0WE_	P10 U	I/O	4	Addr bit 26	Logical AND of CS0_ and WE_
External bus					**External bus**	
ADDR25		M10 U	I/O	4	Remainder of address bus (through ADDR0)	
ADDR24		R10 U	I/O	4		
ADDR23		N9 U	I/O	4		
ADDR22		R9 U	I/O	4		
ADDR21		M9 U	I/O	4		
ADDR20		N8 U	I/O	4		
ADDR19		P8 U	I/O	4		
ADDR18		M7 U	I/O	4		
ADDR17		R7U	I/O	4		
ADDR16		N7 U	I/O	4		
ADDR15		R6 U	I/O	4		
ADDR14		M6 U	I/O	4		
ADDR13		P6 U	I/O	4		
ADDR12		N6 U	I/O	4		
ADDR11		M5 U	I/O	4		
ADDR10		P5 U	I/O	4		
ADDR9		N5 U	I/O	4		
ADDR8		R4 U	I/O	4		

Table 2: System bus interface pinout

Symbol	Pin	I/O	OD	Description
ADDR7	R3 U	I/O	4	
ADDR6	R2 U	I/O	4	
ADDR5	M4 U	I/O	4	
ADDR4	N4 U	I/O	4	
ADDR3	R1 U	I/O	4	
ADDR2	M3 U	I/O	4	
ADDR1	N2 U	I/O	4	
ADDR0	P1 U	I/O	4	
DATA31	N1	I/O	4	Data bus
DATA30	M1	I/O	4	
DATA29	L3	I/O	4	
DATA28	L2	I/O	4	
DATA27	L4	I/O	4	
DATA26	L1	I/O	4	
DATA25	K3	I/O	4	
DATA24	K2	I/O	4	
DATA23	K1	I/O	4	
DATA22	J2	I/O	4	
DATA21	J3	I/O	4	
DATA20	J1	I/O	4	
DATA19	H3	I/O	4	
DATA18	H4	I/O	4	
DATA17	H1	I/O	4	
DATA16	H2	I/O	4	
DATA15	G4	I/O	4	
DATA14	G1	I/O	4	
DATA13	G3	I/O	4	

Table 2: System bus interface pinout

Symbol	Pin	I/O	OD	Description
DATA12	G2	I/O	4	
DATA11	F4	I/O	4	
DATA10	F2	I/O	4	
DATA9	F3	I/O	4	
DATA8	E1	I/O	4	
DATA7	E2	I/O	4	
DATA6	E3	I/O	4	
DATA5	D1	I/O	4	
DATA4	C1	I/O	4	
DATA3	B1	I/O	4	
DATA2	D4	I/O	4	
DATA1	D3	I/O	4	
DATA0	C2	I/O	4	
BE3_	D9	I/O	2	Byte enable D31:D24
BE2_	A9	I/O	2	Byte enable D23:D16
BE1_	C9	I/O	2	Byte enable D15:D08
BE0_	B9	I/O	2	Byte enable D07:D00
TS_	A8 U	I/O	4	DO NOT USE Add an external 820 ohm pullup to 3.3 V.
TA_	D8 U	I/O	4	Data transfer acknowledge Add an external 820 ohm pullup to 3.3 V. TA_ is bidirectional. It is used in input mode to terminate a memory cycle externally. It is used in output mode for reference purposes only.

Table 2: System bus interface pinout

Symbol	Pin	I/O	OD	Description
TEA_	C8 U	I/O	4	Data transfer error acknowledge Add an external 820 ohm pullup to 3.3 V. TEA_ is bidirectional. It is used in input mode to terminate a memory cycle externally. It is used in output mode for reference purposes only.
RW_	D6	I/O	2	Transfer direction
BR_	D7	NO CONNECT		
BG_	C7	NO CONNECT		
BUSY_	B7	NO CONNECT		

Table 2: System bus interface pinout

Signal descriptions

Mnemonic	Signal	Description
BCLK	Bus clock	Provides the bus clock. All system bus interface signals are referenced to the BCLK signal.
ADDR[27:0]	Address bus	Identifies the address of the peripheral being addressed by the current bus master. The address bus is bi-directional.
DATA[31:0]	Data bus	Provides the data transfer path between the NS7520 and external peripheral devices. The data bus is bi-directional. **Recommendation:** Less than x32 (S)DRAM/SRAM memory configurations. Unconnected data bus pins will float during memory read cycles. Floating inputs can be a source of wasted power. For other than x32 DRAM/SRAM configurations, the unused data bus signals should be pulled up.
TS_	Transfer start	NO CONNECT
BE_	Byte enable	Identifies which 8-bit bytes of the 32-bit data bus are active during any given system bus memory cycle. The BE_ signals are active low and bi-directional.

Table 3: System bus interface signal description

Mnemonic	Signal	Description
TA_	Transfer acknowledge	Indicates the end of the current system bus memory cycle. This signal is driven to 1 prior to tri-stating its driver. TA_ is bi-directional.
TEA_	Transfer error acknowledge	Indicates an error termination or burst cycle termination: ■ In conjunction with TA_ to signal the end of a burst cycle. ■ Independently of TA_ to signal that an error occurred during the current bus cycle. TEA_ terminates the current burst cycle. This signal is driven to 1 prior to tri-stating its driver. TEA_ is bi-directional. The NS7520 or the external peripheral can drive this signal.
RW_	Read/write indicator	Indicates the direction of the system bus memory cycle. RW_ high indicates a read operation; RW_ low indicates a write operation. The RW_ signal is bi-directional.
BR_	Bus request	NO CONNECT
BG_	Bus grant	NO CONNECT
BUSY_	Bus busy	NO CONNECT

Table 3: System bus interface signal description

Chip select controller

The NS7520 supports five unique chip select configurations:

Symbol	Pin	I/O	OD	Description
CS4_	B4	O	4	Chip select/DRAM RAS_
CS3_	A4	O	4	Chip select/DRAM RAS_
CS2_	C5	O	4	Chip select/DRAM RAS_
CS1_	B5	O	4	Chip select/DRAM RAS_

Table 4: Chip select controller pinout

Symbol	Pin	I/O	OD	Description
CS0_	D5	O	4	Chip select (boot select)
CAS3_	A1	O	4	FP/EDO DRAM column strobe D31:D24/SDRAM RAS_
CAS2_	C4	O	4	FP/EDO DRAM column strobe D23:D16/SDRAM CAS_
CAS1_	B3	O	4	FP/EDO DRAM column strobe D15:D08/SDRAM WE_
CAS0_	A2	O	4	FP/EDO DRAM column strobe D07:D00/SDRAM A10(AP)
WE_	C6	O	4	Write enable for NCC Ctrl'd cycles
OE_	B6	O	4	Output enable for NCC Ctrl'd cycles

Table 4: Chip select controller pinout

Signal descriptions

Mnemonic	Signal	Description
CS0_ CS1_ CS2_ CS3_ CS4_	Chip select 0 Chip select 1 Chip select 2 Chip select 3 Chip select 4	Unique chip select outputs supported by the NS7520. Each chip select can be configured to decode a portion of the available address space and can address a maximum of 256 Mbytes of address space. The chip selects are configured using registers in the memory module. A chip select signal is driven low to indicate the end of the current memory cycle. For FP/EDO DRAM, these signals provide the RAS signal.

Table 5: Chip select controller signal description

Mnemonic	Signal	Description
CAS0_ CAS1_ CAS2_ CAS3_	Column address strobe signals	Activated when an address is decoded by a chip select module configured for DRAM mode. The CAS_ signals are active low and provide the column address strobe function for DRAM devices. The CAS_ signals also identify which 8-bit bytes of the 32-bit data bus are active during any given system bus memory cycle. For SDRAM, CAS[3:1]_ provides the SDRAM command field. CAS0_ provides the auto-precharge signal. For non-DRAM settings, these signals are 1.
WE_	Write enable	Active low signal that indicates that a memory write cycle is in progress. This signal is activated only during write cycles to peripherals controlled by one of the chip selects in the memory module.
OE_	Output enable	Active low signal that indicates that a memory read cycle is in progress. This signal is activated only during read cycles from peripherals controlled by one of the chip selects in the memory module.

Table 5: Chip select controller signal description

Ethernet interface MAC

Note: ENDEC values for general-purpose output and TXD refer to bits in the Ethernet General Control register. ENDEC values for general-purpose input and RXD refer to bits in the Ethernet General Status register.

In this table, *GP* designates *general-purpose*.

Symbol		Pin	I/O	OD	Description	
MII	**ENDEC**				**MII**	**ENDEC**
MDC	GP output	D10	O	2	MII management clock	State of (LPBK bit XOR (Mode = SEEQ))
MDIO	GP output	B10 U	I/O	2	MII data	State of UTP_STP bit
TXCLK		C10	I		TX clock	

Table 6: Ethernet interface MAC pinout

Symbol		Pin	I/O	OD	Description	
TXD3	GP output	A12	O	2	TX data 3	State of AUI_TP[0] bit
TXD2	GP output	B11	O	2	TX data 2	State of AUI_TP[1] bit
TXD1	GP output	D11	O	2	TX data 1	Inverted state of PDN bit, open collector
TXD0	TXD	A11	O	2	TX data 0	Transmit data
TXER	GP output	A13	O	2	TX code error	State of LNK_DIS_ bit
TXEN		B12	O	2	TX enable	
TXCOL		A14	I		Collision	
RXCRS		D12	I		Carrier sense	
RXCLK		C12	I		RX clock	
RXD3	GP input	D14	I		RX data 3	Read state in bit 12
RXD2	GP input	B15	I		RX data 2	Read state in bit 15
RXD1	GP input	A15	I		RX data 1	Read state in bit 13
RXD0	RXD	B13	I		RX data 0	Receive data
RXER	GP input	C15	I		RX error	Read state in bit 11
RXDV	GP input	D15	I		RX data valid	Read state in bit 10

Table 6: Ethernet interface MAC pinout

Signal descriptions

The Ethernet MII (media independent interface) provides the connection between the Ethernet PHY and the MAC (media access controller).

Mnemonic	Signal	Description
MDC	MII management clock	Provides the clock for the MDIO serial data channel. The MDC signal is an NS7520 output. The frequency is derived from the system operating frequency per the CLKS field setting (see the CLKS field in Table 69: "MII Management Configuration register bit definition" on page 191).

Table 7: Ethernet interface MAC signal description

Mnemonic	Signal	Description
MDIO	Management data IO	A bi-directional signal that provides a serial data channel between the NS7520 and the external Ethernet PHY module.
TXCLK	Transmit clock	An input to the NS7520 from the external PHY module. TXCLK provides the synchronous data clock for transmit data.
TXD3 TXD2 TXD1 TXD0	Transmit data signals	Nibble bus used by the NS7520 to drive data to the external Ethernet PHY. All transmit data signals are synchronized to TXCLK. In ENDEC mode, only TXD0 is used for transmit data.
TXER	Transmit coding error	Output asserted by the NS7520 when an error has occurred in the transmit data stream.
TXEN	Transmit enable	Asserted when the NS7520 drives valid data on the TXD outputs. This signal is synchronized to TXCLK.
COL	Transmit collision	Input signal asserted by the external Ethernet PHY when a collision is detected.
CRS	Receive carrier sense	Asserted by the external Ethernet PHY whenever the receive medium is non-idle.
RXCLK	Receive clock	An input to the NS7520 from the external PHY module. The receive clock provides the synchronous data clock for receive data.
RXD3 RXD2 RXD1 RXD0	Receive data signals	Nibble bus used by the NS7520 to input receive data from the external Ethernet PHY. All receive data signals are synchronized to RXCLK. In ENDEC mode, only RXD0 is used for receive data.
RXER	Receive error	Input asserted by the external Ethernet PHY when the Ethernet PHY encounters invalid symbols from the network.
RXDV	Receive data valid	Input asserted by the external Ethernet PHY when the PHY drives valid data on the RXD inputs.

Table 7: Ethernet interface MAC signal description

"No connect" pins

Pin	Description
R13	Tie to V_{CC}
P12	Tie to V_{CC}
N12	XTALB1: Tie to V_{CC}
R15	XTALB2: NO CONNECT
M11	NO CONNECT
P11	NO CONNECT
N11	NO CONNECT
R12	NO CONNECT
R14	NO CONNECT
P13	NO CONNECT

Table 8: "No connect" pins

General-purpose I/O

GPIO signal	Serial signal	Other signal	Pin	I/O	OD	Serial channel description	Other description
PORTA7	TXDA		J14 U	I/O	2	Channel 1 TXD	
PORTA6	DTRA_	DREQ1_	J13 U	I/O	2	Channel 1 DTR_	DMA channel 3/5 Req
PORTA5	RTSA_		J15 U	I/O	2	Channel 1 RTS_	
PORTA4	RXCA/ RIA_/ OUT1A_		J12 U	I/O	2	Pgm'able Out/ Channel 1 RXCLK/ Channel 1 ring signal/Channel 1 SPI clock (CLK)	

Table 9: GPIO pinout

GPIO signal	Serial signal	Other signal	Pin	I/O	OD	Serial channel description	Other description
PORTA3	RXDA	DACK1_	H15 U	I/O	2	Channel 1 RXD	DMA channel 3/5 ACK
PORTA2	DSRA_	AMUX	H12 U	I/O	2	Channel 1 DSR_	DRAM addr mux
PORTA1	CTSA_	DONE1_ (O)	H13 U	I/O	2	Channel 1 CTS_	DMA channel 3/5 DONE_Out
PORTA0	TXCA/ OUT2A_/ DCDA_	DONE1_ (I)	G12 U	I/O	2	Pgm'able Out/ Channel 1 DCD/Channel 1 SPI enable (SEL_)/Channel 1 TXCLK	DMA channel 3/5 DONE_In
PORTC7	TXDB		G13 U	I/O	2	Channel 2 TXD	GEN interrupt out
PORTC6	DTRB_	DREQ2_	G14 U	I/O	2	Channel 2 DTR_	DMA channel 4/6 Req
PORTC5	RTSB_	REJECT_	F15 U	I/O	2	Channel 2 RTS_	CAM reject
PORTC4	RXCB/RIB_/ OUT1B_	RESET_	F12 U	I/O	2	Pgm'able Out/ Channel 2 RXCLK/ Channel 2 ring signal/Channel 2 SPI clock (CLK)	RESET output See Note 1 following this table.
PORTC3[2]	RXDB	LIRQ3/ DACK2_	F13 U	I/O	2	Channel 2 RXD	Level sensitive IRQ/DMA channel 4/6 ACK
PORTC2[2]	DSRB_	LIRQ2/ RSPF_	E15 U	I/O	2	Channel 2 DSR_	Level sensitive IRQ/CAM request
PORTC1[2]	CTSB_	LIRQ1/ DONE2_ (O)	E12 U	I/O	2	Channel 2 CTS_	Level sensitive IRQ/DMA channel 4/6 DONE_out

Table 9: GPIO pinout

GPIO signal	Serial signal	Other signal	Pin	I/O	OD	Serial channel description	Other description
PORTC0[2]	TXCB/ OUT2B_/ DCDB_	LIRQ0/ DONE2_(I)	E14 U	I/O	2	Pgm'able Out/ Channel 2 DCD/Channel 2 SPI enable (SEL_)/Channel 2 TXCLK	Level sensitive IRQ/DMA channel 4/6 DONE_in

Table 9: GPIO pinout

Notes:

1 RESET output indicates the reset state of the NS7520. PORTC4 persists beyond the negation of RESET_ for approximately 512 clock cycles if the PLL is disabled. When the PLL is enabled, PORTC4 persists beyond the negation of RESET_ to allow for PLL lock for 100 microseconds times the ratio of the VCO to XTALA.

Note that this GPIO is left in output mode active following a hardware RESET.

2 *PORTC[3:0] pins provide level-sensitive interrupts. The inputs do not need to be synchronous to any clock. The interrupt remains active until cleared by a change in the input signal level.

Signal descriptions

See Chapter 6, "GEN Module," for signal and configuration information for PORTA and PORTC.

System clock and reset

Symbol	Pin	I/O	OD	Description
XTALA1	K14	I		ARM/system oscillator circuit
XTALA2	K12	O		
PLLVDD (1.5V)	L15	P		PLL clean power
PLLVSS	L12	P		PLL return
RESET_	A10	I		System reset

Table 10: System clock pinout

Signal descriptions

The NS7520 has three clock domains:

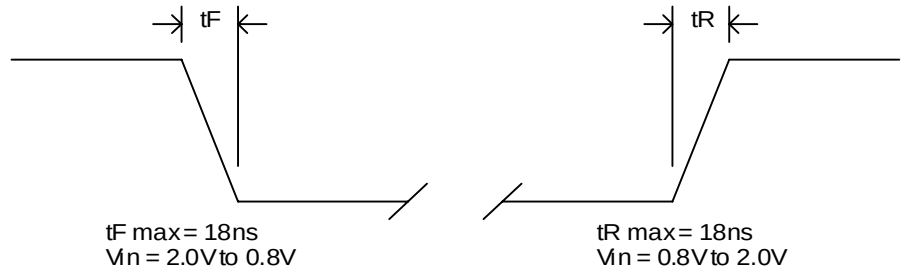
- System clock (SYSCLK)
- Bit rate generation and programmable timer reference clock (XTALA1/2)
- System bus clock (BCLK)

The SYS module provides the NS7520 with these clocks, as well as system reset and backup resources.

Mnemonic	Signal	Description
XTALA1 XTALA2	Oscillator input Oscillator output	A standard parallel quartz crystal or crystal oscillator can be attached to these pins to provide the main input clock to the NS7520.
PLLVDD PLLVSS	Clean PLL power Connect directly to the GND plane	Power and ground for PLL circuit.
RESET_	System reset	Resets the NS7520 hardware.

Table 11: Clock generation and reset signal description

This figure shows the timing and specification for RESET_ rise/fall times:



System mode (test support)

PLLTST_, BISTEN_, and SCANEN_ primary inputs control different test modes for both functional and manufacturing test operations. See Chapter 5, "SYS Module," for more information.

Symbol	Pin	I/O	OD	Description
PLLTST_	N15	I		Encoded with BISTEN_ and SCANEN_ Add an external pullup to 3.3V or pulldown to GND.
BISTEN_	M15	I		Encoded with PLLTST_ and SCANEN_ Add an external pullup to 3.3V or pulldown to GND.
SCANEN_	L13	I		Encoded with BISTEN_ and PLLTST_ Add an external pullup to 3.3V or pulldown to GND.

Table 12: System mode and system reset pinout

JTAG test (ARM debugger)

JTAG boundary scan allows a tester to check the soldering of all signal pins and tri-state all outputs.

Symbol	Pin	I/O	OD	Description
TDI	N14 U	I		Test data in
TDO	M13	O	2	Test data out
TMS	M12 U	I		Test mode select
TRST_	M14	I		Test mode reset Requires external termination when not being used (see Figure 4, "TRST_ termination," on page 27 for an illustration of the termination circuit on the development PCB).
TCK	P15	I		Test mode clock Add an external pullup to 3.3V.

Table 13: JTAG test pinout

Signal descriptions

Mnemonic	Signal	Description
TDI	Test data in	TDI operates the JTAG standard. Consult the JTAG specifications for use in boundary-scan testing. These signals meet the requirements of the Raven and Jeeni debuggers.
TDO	Test data out	TDO operates the JTAG standard. Consult the JTAG specifications for use in boundary-scan testing. These signals meet the requirements of the Raven and Jeeni debuggers.
TMS	Test mode select	TMS operates the JTAG standard. Consult the JTAG specifications for use in boundary-scan testing. These signals meet the requirements of the Raven and Jeeni debuggers.

Table 14: ARM debugger signal description

Mnemonic	Signal	Description
TRST_	Test mode reset	TRST_ operates the JTAG standard. Consult the JTAG specifications for use in boundary-scan testing. These signals meet the requirements of the Raven and Jeeni debuggers.
TCK	Test mode clock	TCK operates the JTAG standard. Consult the JTAG specifications for use in boundary-scan testing. These signals meet the requirements of the Raven and Jeeni debuggers.

Table 14: ARM debugger signal description

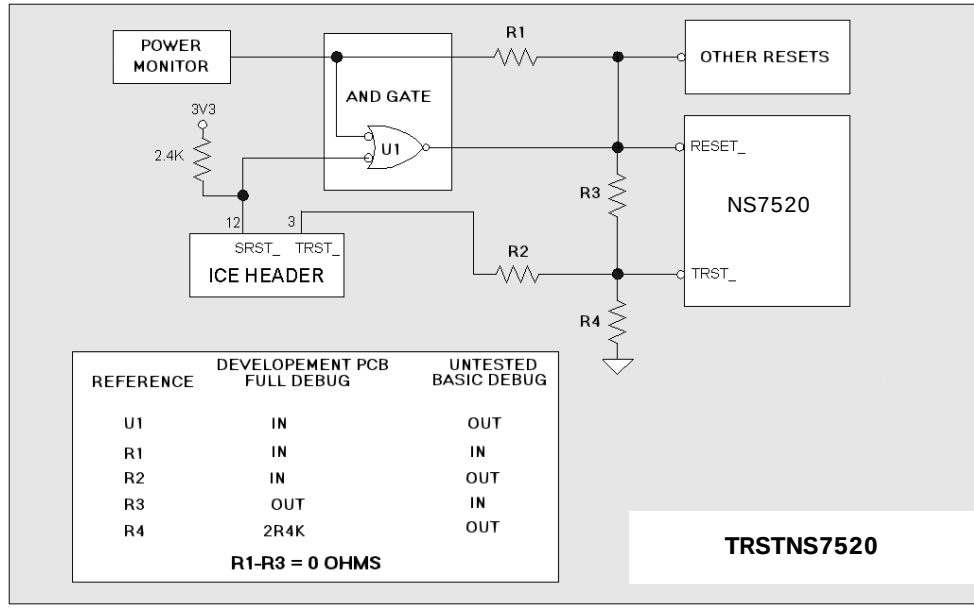


Figure 4: TRST_ termination

Power supply

Signal	Pin	Description
Oscillator VCC (3.3V)	N13, C3	Oscillator power supply
Core VCC (1.5V)	R8, L14, C14, C13	Core power supply
I/O VCC (3.3V)	E4, K4, M2, N3, P3, R5, H14, F14, B8, A3	I/O power supply
GND	D2, F1, J4, P4, P7, M8, P9, R11, K15, G15, E13, D13, B14, C11, A7, A5, B2, P2, P14, K13	Ground

Table 15: Power supply pinout

See the DC and AC electrical specifications in Chapter 11, "Electrical Characteristics," for more information.

Working with the CPU

CHAPTER 3

The CPU uses an ARM7TDMI core processor, which provides high performance while maintaining low power consumption and small size. This chapter describes the ARM Thumb concept and provides an overview of ARM exceptions and hardware interrupts.

ARM Thumb concept

The ARM7TDMI processor uses a unique architectural strategy known as *Thumb*, which makes the processor ideally suited to high-volume applications with memory restrictions or applications for which code density is an issue.

Thumb code's primary attribute is a super-reduced instruction set. The ARM7TDMI processor has essentially two instruction sets:

- Standard 32-bit ARM set
- 16-bit Thumb set

Thumb's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. Thumb code operates on the same 32-bit register set as the ARM code, but consumes only 65% of the same code compiled in ARM mode.

Thumb instructions operate with the standard ARM register configuration, allowing interoperability between ARM and Thumb states. Each 16-bit Thumb instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model.

Thumb architecture provides a Thumb instruction decoder in front of the standard 32-bit ARM processor. The Thumb instruction decoder basically remaps each 16-bit Thumb instruction into a 32-bit standard ARM instruction. The Thumb instruction set typically requires 30% more instructions to perform the same task as 32-bit instructions, but the Thumb instruction can fit twice as many instructions in the code space. The net result is a 35% decrease in overall code density.

CPU performance

The ARM7TDMI core does not contain cache, and runs as fast as instructions can be fetched. The performance rating for the ARM RISC depends on system bus speed and cycle time. Performance is also affected by the size of the system bus and the type of code (ARM or Thumb) being executed.

The ARM instruction set yields a 0.9 Dhrystone (2.1) rating MIPS/MHz of instruction executions; the Thumb instruction set yields 0.75 Dhrystones MIPS/MHz. The MHz rating reflects the rate at which instructions can be fetched from external flash memory, as shown in this table:

System bus size	Code style	RISC speed	System bus speed	Wait states	Instruction cycle time	Dhrystone rating
Thumb mode						
16-bit	Thumb	25 MHz	25 MHz	1	120 ns	N/A
16-bit	Thumb	25 MHz	25 MHz	0	80 ns	N/A
16-bit	Thumb	36 MHz	36 MHz	3	125 ns	N/A
16-bit	Thumb	46 MHz	46 MHz	4	109 ns	N/A
16-bit	Thumb	55 MHz	55 MHz	5	108 ns	N/A
ARM mode						
32-bit	ARM	36 MHz	36 MHz	3	125 ns	6.8
32-bit	ARM	46 MHz	46 MHz	4	109 ns	8.6
32-bit	ARM	55 MHz	55 MHz	5	108 ns	10.4

Table 16: ARM performance

Working with ARM exceptions

Exceptions occur when the normal flow of a program is halted temporarily; for example, to service an interrupt from a peripheral. Each ARM exception causes the ARM processor to save some state information, then jump to a location in low memory (referred to as the *vector table*; see "Exception vector table" on page 33).

Before an exception can be handled, the current processor state must be preserved so the original program can resume when the handler routine has finished.

Summary of ARM exceptions

The ARM processor can be interrupted by any of seven basic exceptions:

- **Reset exception.** After a reset condition, the ARM7TDMI saves the current values of the PC (program counter) and CPSR (Current Processor Status register).
- **Undefined exception.** The ARM7TDMI takes the undefined instruction trap when it finds an instruction it cannot handle.
- **SWI instruction.** The ARM7TDMI uses the *software interrupt instruction* (SWI) to enter supervisor mode, usually to request a specific supervisor instruction.
- **Abort exception.** An abort exception indicates that the current memory access cannot be completed. There are two types of abort exception:
  - **Prefetch.** Occurs during an instruction prefetch.
  - **Data.** Occurs during a data operand access.
- **IRQ.** An *interrupt request* (IRQ) exception is a normal interrupt serviced by the ARM7TDMI controller.
- **FIRQ.** A *fast interrupt request* (FIRQ) exception supports a data transfer or channel process. An FIRQ interrupt is generated only by the GEN module timers and watchdog timer.

Exception priorities

Several exceptions can occur at the same time. If this happens, a fixed-priority system determines the order in which they are handled:

Highest priority

1 Reset

2 Data abort

3 FIRQ

4 IRQ

5 Prefetch abort

6 Undefined instruction, SWI

Lowest priority

Not all exceptions can occur at the same time, however.

- Undefined instructions and SWIs are mutually exclusive, as they each correspond to particular (non-overlapping) decoding of the current instruction.
- If a data abort occurs at the same time as FIRQ and the FIRQ is enabled (that is, the CPSR F flag is clear), the data abort takes priority. ARM7TDMI enters the data abort handler and immediately goes to the FIRQ vector. A normal return from FIRQ causes the data abort handler to resume execution.

  Placing data abort at a higher priority than FIRQ is necessary to ensure that the transfer error does not escape detection. The time for this exception entry should be added to worst-case FIRQ latency calculations.

Exception vector table

All exceptions result in the ARM processor vectoring to an address in low memory, using the exception vector table. The exception vector table always exists and always starts at base address 0.

Vector address	Vector	Description
'h0	RESET	Reset vector; for initialization and startup
'h4	Undefined	Undefined instruction encountered
'h8	SWI	Software interrupt; used for entry point into the kernel
'hC	Abort (prefetch)	Bus error (no response or error) fetching instructions
'h10	Abort (data)	Bus error (no response or error) fetching data
'h14	Reserved	Reserved
'h18	IRQ	Interrupt from ARM7TDMI interrupt controller
'h1C	FIRQ	Fast interrupt from ARM7TDMI controller

Table 17: Exception vector table

All internal ARM7TDMI internal peripherals are presented to the CPU using the IRQ or FIRQ interrupt inputs. The ARM can mask various ARM7TDMI peripheral interrupts at the global level, using the ARM7TDMI interrupt controller. The ARM also can mask interrupts at the micro-level, using configuration features with the peripheral modules.

All IRQ interrupts are disabled when the *I* bit is set in the ARM CPSR. When the *I* bit is cleared, those interrupts enabled in the ARM7TDMI interrupt controller can assert the IRQ input to the ARM processor.

The ARM processor sets the I bit automatically when entering an interrupt service routine (ISR), which disables recursive interrupts. The ISR's first task is to read the Interrupt Status register, which identifies all active sources for the IRQ interrupt. Firmware sets the priorities for servicing interrupts at bootup, using the bits defined in the Interrupt Status register.

Detail of ARM exceptions

Reset exception

A reset exception is the highest priority exception. When the ARM7TDMI is held in reset, the processor abandons the executing instruction and continues to fetch instructions from incrementing word addresses.

When the ARM7TDMI is removed from reset, the processor performs these steps:

1. Overwrites `R14_svc` and `SPSR_svc` (Saved Processor Status register) by copying the current values of the PC and CPSR into them. The values of the saved PC and SPSR are not defined.
2. Forces the CPSR M field to `10011` (supervisor mode), sets the I and F bits in the CPSR, and clears the CPSR T bit (back to ARM mode).
3. Forces the PC to fetch the next instruction from address `'h00`.
4. Resumes execution in ARM state.

Undefined exception

When the ARM7TDMI encounters an instruction it cannot handle, it takes the undefined instruction trap. The undefined instruction trap can extend either the Thumb or ARM instruction set by software emulation.

After emulating the failed instruction, the trap handler should execute the following instruction irrespective of the state (Thumb or ARM): `MOVS PC, R14_und`.

This instruction restores the PC and CPSR, and returns to the instruction following the undefined instruction.

SWI exception

An SWI is used for entering supervisor mode, usually to request a particular supervisor function. An SWI handler should return by executing this instruction irrespective of the state (ARM or Thumb): `MOVS PC, R14_SVC`.

This instruction restores the PC and CPSR, and returns to the instruction following the SWI.

Abort exception

An abort indicates that the current memory access cannot be completed, and is signaled by the external ABORT input. The ARM7TDMI checks for the abort exception during memory access cycles.

There are two types of abort exception:

- **Prefetch abort.** Occurs during an instruction prefetch. If a prefetch abort occurs, the prefetch instruction is marked as invalid but the exception is not taken until the instruction reaches the head of the pipeline. If the instruction is not executed (for example, if a branch occurs while the instruction is in the pipeline), the abort does not take place.
- **Data abort.** Occurs during a data operand access. If a data abort occurs, the action taken depends on the instruction type:
  - Single data transfer instructions (LDR, STR) write back modified base registers; the abort handler must be aware of this.
  - A swap instruction (SWP) is aborted as though it had not been executed.
  - Block data transfer instructions (LDM, STM) complete. If write-back is set, the base is updated. If the instruction would have overwritten the base with data (that is, the base is in the transfer list), the overwriting is prevented. All register overwriting is prevented after an abort is indicated, which means that `R15` (always the last register to be transferred) is preserved in an aborted LDM instruction.

The abort mechanism allows the implementation of a demand-paged virtual memory system. In this type of system, the processor is allowed to generate arbitrary addresses. When the data at an address is unavailable, the memory management unit (MMU) signals an abort. The abort handler must then work out the cause of the abort, make the requested data available, and retry the aborted instruction. The application program needs no knowledge of the amount of memory available to it, and its state is not affected by the abort.

The handler executes one of the following instructions, irrespective of the state (ARM or Thumb), after fixing the cause of the abort:

- For a prefetch abort: `SUBS PC, R14_abt, #4`
- For a data abort: `SUBS PC, R14_abt, #8`

IRQ exception

An IRQ exception is a normal interrupt sourced by the ARM7TDMI interrupt controller. IRQ has a lower priority than FIRQ, and is masked out when an FIRQ sequence is entered. IRQ can be disabled at any time by setting the I bit in CPSR to 1; this can be done only from privileged (non-user) mode.

The IRQ handler should leave the interrupt by executing the following instruction irrespective of the state (ARM or Thumb): `SUBS PC, R14_irq, #4`.

FIRQ exception

An FIRQ exception supports a data transfer or channel process. In ARM state, FIRQ has enough registers to remove the need for register saving, which minimizes context switching overhead.

Only two peripherals can generate an FIRQ interrupt: the GEN module built-in timers and the GEN module watchdog timer.

The FIRQ handler should leave the interrupt by executing the following instruction irrespective of the state (ARM or THUMB): `SUBS PC, R14_firq, #4`.

The FIRQ interrupt can be disabled by setting the CPSR F flag to 1, only in non-user mode. If the F flag is clear, the ARM7TDMI checks for a low level on the output of the FIRQ synchronizer at the end of each instruction.

Entering and exiting an exception (software action)

The ARM7TDMI performs specific steps when handling exceptions.

Entering an exception

When handling an exception, ARM7TDMI does this:

1 Preserves the address of the next instruction in the appropriate Link register.

- If the exception has been entered from the ARM state, the address of the next instruction is copied into the Link register. The address is either current `PC + 4` or `PC + 8`, depending on the exception. (See Table 18: "Exception entry/exit by exception type" on page 38.)
- If the exception has been entered from Thumb state, the value written into the Link register is the current PC offset by a value that lets the program continue from the correct place on return from the exception.

The exception handler does not need to determine from which state the exception was entered. With an SWI, for example, `MOVS PC, R14_SVC` always returns to the next instruction whether executed in ARM or Thumb state.

2 Copies the CPSR into the appropriate SPSR.

3 Forces the CPSR mode bits to a value that depends on the exception.

4 Forces the PC to fetch the next instruction from the relevant exception vector.

5 Sets the I (for IRQ interrupts) or F (for FIRQ interrupts) bits to disable interrupts to prevent unmanageable nesting of exceptions.

Note: If the processor is in Thumb state when an exception occurs, it automatically switches into ARM state when the PC is loaded with the exception vector address.

Exiting an exception

On completion, ARM7TDMI does this:

1 Moves the Link register, minus the offset where appropriate, to the PC. The offset value varies depending on the type of exception.

2 Copies the SPSR back to the CPSR.

3 Clears the interrupt disable flags, if they were set on entry.

Note: An explicit switch back to Thumb state is never needed. Restoring the CPSR from the SPSR automatically sets the T bit to the value it held immediately before the exception.

Exception entry/exit summary

In the variable `R14_x`, `R14` is the Link register; `_x` is the previous state of the processor.

Return/ exception	Return instruction	Previous state ARM R14_x	Previous state Thumb R14_x	Notes
BL	`MOV PC, R14`	PC + 4	PC + 2	1
RESET	NA	NA	NA	4
UNDEF	`MOVS PC, R14_und`	PC + 4	PC + 2	1
SWI	`MOVS PC, R14_svc`	PC + 4	PC + 2	1
ABORT P	`SUBS PC, R14_abt, #4`	PC + 4	PC + 4	1
ABORT D	`SUBS PC, R14_abt, #8`	PC + 8	PC + 8	3
IRQ	`SUBS PC, R14_irq, #4`	PC + 4	PC + 4	2
FIRQ	`SUBS PC, R14_firq, #4`	PC + 4	PC + 4	2

Table 18: Exception entry/exit by exception type

Notes:

1 Where PC is the address of the BL/SWI/undefined instruction fetch that had the prefetch abort. *BL* is a *branch with link* instruction.

2 Where PC is the address of the instruction that was not executed since FIRQ or IRQ took priority.

3 Where PC is the address of the load or store instruction that generated the data abort.

4 The value saved in `R14_svc` upon reset is unpredictable.

Hardware Interrupts

Two wires that go into the ARM7 CPU core can interrupt the processor:

- IRQ (normal interrupt)
- FIRQ (fast interrupt)

Although the interrupts are basically the same, FIRQ can interrupt IRQ.

FIRQ and IRQ lines

The FIRQ line adds a simple, two-tier priority scheme to the interrupt system. Most sources of interrupts on the ARM7TDMI come from the IRQ line. The only potential sources for FIRQ interrupts in the ARM7TDMI come from the two built-in timers and the watchdog timer; there is no way to generate an FIRQ signal externally. These timers are controlled by registers in the GEN module (see "Timer Control registers," beginning on page 70):

- The built-in timers are controlled using the Timer Control registers (`'hFFB0 0010/18`). The corresponding bit in the Interrupt Enable register must be set for either IRQ or FIRQ to function.
- The watchdog timer is controlled using the System Control register (`'hFFB0 0000`).

Interrupt controller

Interrupts come from many different sources on the ARM7TDMI, and are managed by the interrupt controller within the GEN module. Interrupts can be enabled or disabled on a per-source basis using the Interrupt Enable register (`'hFFB0 0030`), which serves as a mask for the interrupt sources and ultimately controls whether an interrupt from an ARM7TDMI module can reach the IRQ line.

There are two read-only registers in the interrupt controller:

- **Interrupt Status Register Raw.** Indicates the source of an ARM7TDMI interrupt regardless of the state of the Interrupt Enable register. All interrupts that are active in their respective module will be visible in the Interrupt Status Register Raw.

- **Interrupt Status Register Enabled.** Identifies the current state of all interrupt sources that are enabled. This register is defined by performing a logical AND of the Interrupt Status Register Raw and the Interrupt Enable register. All bits in the Interrupt Status Register Enabled are ORed together. The output is fed directly to the IRQ line, which then interrupts the ARM.

Interrupt sources

Each interrupt source is enabled and disabled within its respective module (and submodule) within the NS7520 ASIC. The interrupt controller in the GEN module, however, does not latch any of the interrupt signals.

Note: Interrupt causes are latched in their respective submodule until cleared.

Interrupt sources include the following:

- **DMA interrupts.** All [13] DMA channels, including the four sub-channels of the Ethernet receiver, have five possible interrupt sources. See Chapter 8, "DMA Module."
- **Ethernet receive and transmit interrupts.** There are three interrupts for Ethernet receive and four interrupts for Ethernet transmit; all interrupts are part of the Ethernet General Status register. These interrupts are used only when the Ethernet receiver and transmitter are in interrupt mode rather than DMA mode. See Chapter 9, "Ethernet Module."
- **Serial interrupts.** The Serial Channel Status register has many interrupt sources. See Chapter 10, "Serial Controller Module."
- **Watchdog timer interrupts.** When the watchdog timer expires, the system can generate either an IRQ interrupt, an FIRQ interrupt, or a system reset. The interrupt type and length of the timer are configured using the GEN module System Control register. The watchdog is strobed using the Software Service register. See Chapter 6, "GEN Module."
- **Timer 1 and Timer 2 interrupts.** Two types of interrupts can be generated by Timers 1 and 2. The interrupt type is configured in the Timer Control register; the interrupt itself is contained within the Timer Status register. See Chapter 6, "GEN Module."

- **PORTC interrupts.** The lower four pins of PORTC (C3, C2, C1, C0) on the ARM7TDMI can be used as interrupt sources. Only the PORTC register enables, configures, and services the interrupts. See Chapter 6, "GEN Module."

BBus Module

C H A P T E R 4

This chapter describes the BBus module, which provides the data path between NS7520 internal modules. Additional BBus functionality includes:

- Address and multiplexing logic that supports the data flow through the NS7520.
- Central arbiter for all NS7520 bus masters.
- Internal register decoding for all addressable modules.

BBus masters and slaves

The BBus module consists of bus master and bus slave modules. The BBus state machine allows each bus master to control the bus in a round-robin manner. If a bus master does not require the bus resources when its turn comes around, that bus is skipped until the next round-robin slot. Each potential bus master presents the BBus with request and attribute signals. Once mastership is granted, the targeted device is selected.

Table 19 illustrates bus master and slave modules.

Module	Master	Slave
CPU module	Y	Y
BUS module	Y	Y
EFE module		Y
DMA module	Y	Y
GEN module		Y
MEM module		Y
SER module		Y

Table 19: BBus masters and slaves

Cycles and BBus arbitration

During a normal cycle, each bus master cycle is allowed only one read/write cycle if another bus master is waiting. There are two exceptions to this rule: burst transactions and read-modify-write transactions.

In a burst transaction, the master can perform more than one read or write cycle. In a read-modify-write transaction, the bus master performs one read and write cycle to the same location.

Address decoding

The CPU address map is divided to allow access to the internal modules and external resources routed through the internal peripherals. Each slave module is given a small portion of the system address map for configuration and status.

Table 20 defines how the address is decoded to allow access to slave modules or the BUS module (external resources).

Address range	Module
0000 0000–FF6F FFFF	BUS module and external memory
FF80 0000–FF8F FFFF	EFE module
FF90 0000–FF9F FFFF	DMA module
FFB0 0000–FFBF FFFF	GEN module
FFC0 0000–FFCF FFFF	MEM module
FFD0 0000–FFDF FFFF	SER module

Table 20: BBus address decoding

All resources defined in this manner are addressed using the upper memory addresses. Each internal module is given 1 Mbyte of address space for its own internal decoding. Each module defines its own specific register map.

The BBus module does not allow access to any internal registers unless the CPU_SUPV signal is active, which indicates that firmware is executing in supervisor mode. The System Control register provides an override signal (the USER bit in the GEN module System Control register) to allow access to internal registers in user mode.

SYS Module

C H A P T E R 5

The SYS module provides the NS7520 with its system clock (SYS_CLK) and system reset (SYS_RESET) resources.

Signal description

The system control signals determine the basic operation of the chip:

Signal mnemonic	Signal name	Description
{XTALA1, XTALA2}	Clock source	Operate in one of two ways: ■ The signals are affixed with a 10-20 MHz parallel mode quartz crystal or crystal oscillator and the appropriate components per the component manufacturer. ■ XTALA1 is driven with a clock signal and XTALA2 is left open.
{PLLVDD, PLLVSS}	PLL power	Provide an isolated power supply for the PLL.
RESET_	Chip reset	Active low signal asserted to initiate a hardware reset of the chip.
{TDI, TDO, TNS, TRST_, TCK}	JTAG interface	Provide a JTAG interface for the chip. This interface is used for both boundary scan and ICE control of the internal processor.
{PLLTEST_, BISTEN_, SCANEN_}	Chip mode	Encoded to determine the chip mode.

JTAG support

The NS7520 provides full support for 1149.1 JTAG boundary scan testing. All NS7520 pins can be controlled using the JTAG interface port. The JTAG interface provides access to the ARM7TDMI debug module when the appropriate combination of PLLTST_, BISTEN_, and SCANEN_ are selected (see "External oscillator mode hardware configuration," beginning on page 51).

ARM debug

The ARM7TDMI core uses a JTAG TAP controller that shares pins with the TAP controller used for 1149.1 JTAG boundary scan testing. To enable the ARM7TDMI TAP controller, {PLLTST_, BISTEN_, and SCANEN_} must be set as shown in "External oscillator mode hardware configuration," beginning on page 51.

System clock generation (NS7520 clock module)

The NS7520 clock module creates the BCLK and FXTAL signals. Both signals are used internally, but BCLK can also be accessed at ball A6 by setting the BCLKD field in the System Control register to 0 (see "System Control register," beginning on page 63).

- BCLK functions as the system clock and provides the majority of the NS7520's timing.
- FXTAL provides the timing for the DRAM refresh counter, can be selected instead of BCLK to provide timing for the watchdog timer, the two internal timers, and the Serial module.

External oscillator vs. internal PLL circuit

The clock module uses either an external oscillator or the internal PLL circuit to produce the BCLK and FXTAL signals. When using an external oscillator, the minimum high/low time on XTALA1 is 4.5ns.

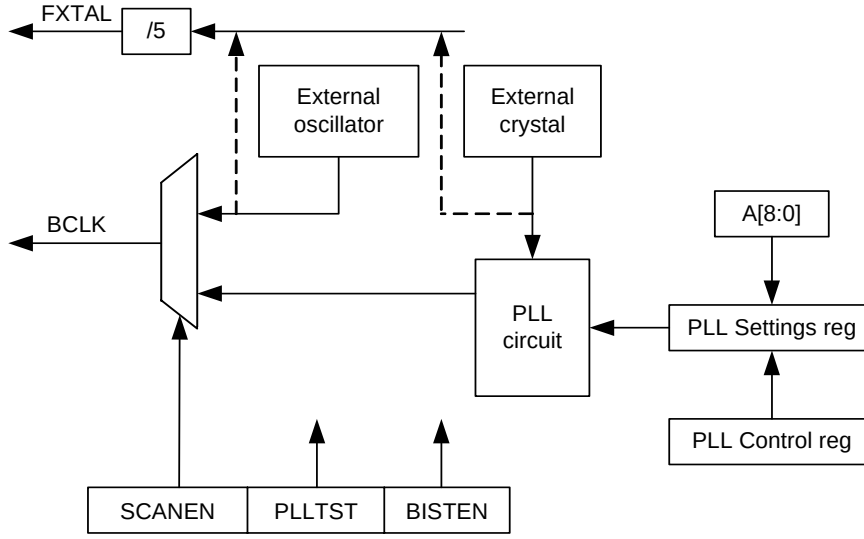
The PLLTST, BISTEN, and SCANEN signals work together to choose between using the external oscillator or the internal PLL circuit in the boundary scan or JTAG debugger modes, as shown:

PLLTST	BISTEN	SCANEN	FUNCTION
0	0	0	N/A
0	0	1	N/A
0	1	0	N/A
0	1	1	External oscillator, boundary scan

PLLTST	BISTEN	SCANEN	FUNCTION
1	0	0	PLL, boundary scan
1	0	1	N/A
1	1	0	PLL, JTAG debugger
1	1	1	External oscillator, JTAG debugger

NS7520 clock module block diagram

This diagram provides an overview of the clock module.



Using the external oscillator

The NS7520 can use an external oscillator to generate the BCLK and FXTAL signals. In this type of configuration, the BCLK frequency is equal to the oscillator input frequency. The FXTAL frequency is equal to the oscillator input frequency divided by five. For example, with a 55MHz oscillator, BCLK would be 55 MHZ and FXTAL would be 11MHz.

Note: Using an external oscillator with PLL enabled is not advantageous, due to the PLL input limitation of 10MHz to 20MHz. The oscillator needs to be the same frequency as the crystal. Using a clock source greater than 20MHz would result in the PLL running outside its operating range.

External oscillator mode hardware configuration

The external oscillator's output is connected to the XTALA1 input through a 100 resistor. XTALA2 is an output and is left open.

The clock module has two power pins: PLLVDD and PLLVSS.

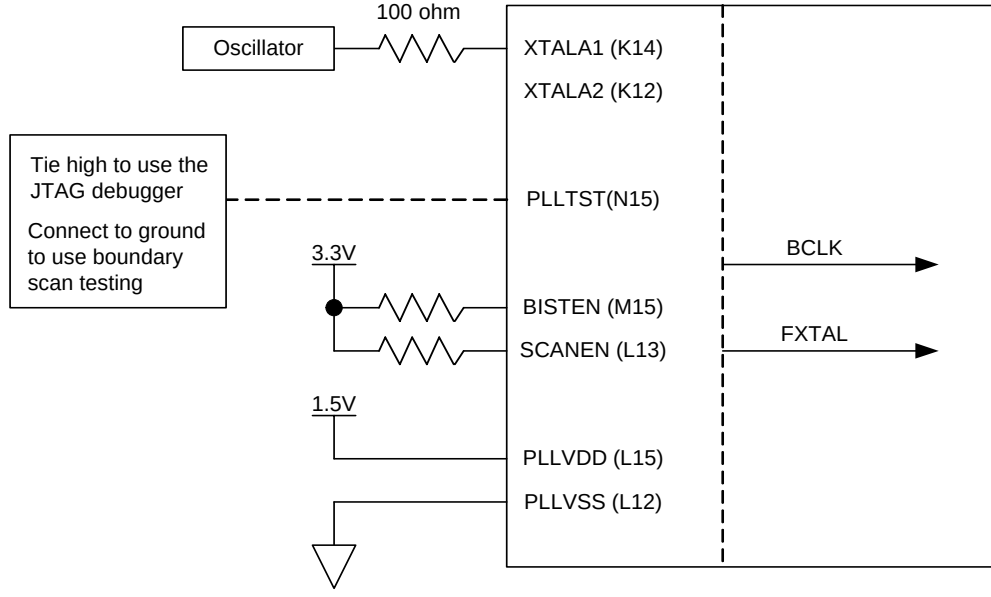
- PLLVDD is connected to 1.5 volts
- PLLVSS is connected to ground

The PLLTST input is connected to ground to use the JTAG debugger. It is connected to 3.3 volts through a 10K resistor to use boundary scan testing.

The BISTEN input is tied to 3.3 volts through a 10K resistor.

The SCANEN input is tied to 3.3 volts through a 10K resistor.

This diagram shows the hardware configuration:



Using the PLL circuit

The NS7520 can use its PLL external oscillator to generate the BCLK and FXTAL signals. In this configuration, BCLK can be set to several values, in increments of 1/4 the crystal frequency. FXTAL us always the crystal frequency divided by five.

PLL mode hardware configuration

Figure 5, "PLL mode hardware configuration," on page 53, shows how the crystal is connected to the XTALA1 and XTALA2 inputs.

- When the clock module is configured to use the PLL, the power to the module must be cleaner than when using an external oscillator.
- PLLVDD must be connected to 1.5V through a ferrite bead and bypassed with a 100nF capacitor placed close to ball L15.
- PLLVSS is connected to ground.
- The PLLTST input i s connected to ground to use the JTAG debugger; it is connected to 3.3V through a 10K resistor to use boundary scan testing.
- The BISTEN* input is tied to 3.3 volts through a 10K resistor.
- The SCANEN* must be low to choose the PLL mode. The SCANEN* input must be connected to system reset through an inverter, to ensure that the PLL circuit is properly reset.
- RESET* must have a *rise time of 18nS from 0.8V to 2.0V.* The MAX811 is in this range. A RESET* source with a slow rise time can be used by using a double inverter. The first inverter's output connects to SCANEN* and to the second inverter input. The second inverter's output drives RESET*.

Address lines A[8:0] configure the PLL circuit on bootup. Note that the values on address lines A8, A6, A5, A4, and A2 are inverted in the PLL Settings register (see "Setting the PLL frequency," beginning on page 54).

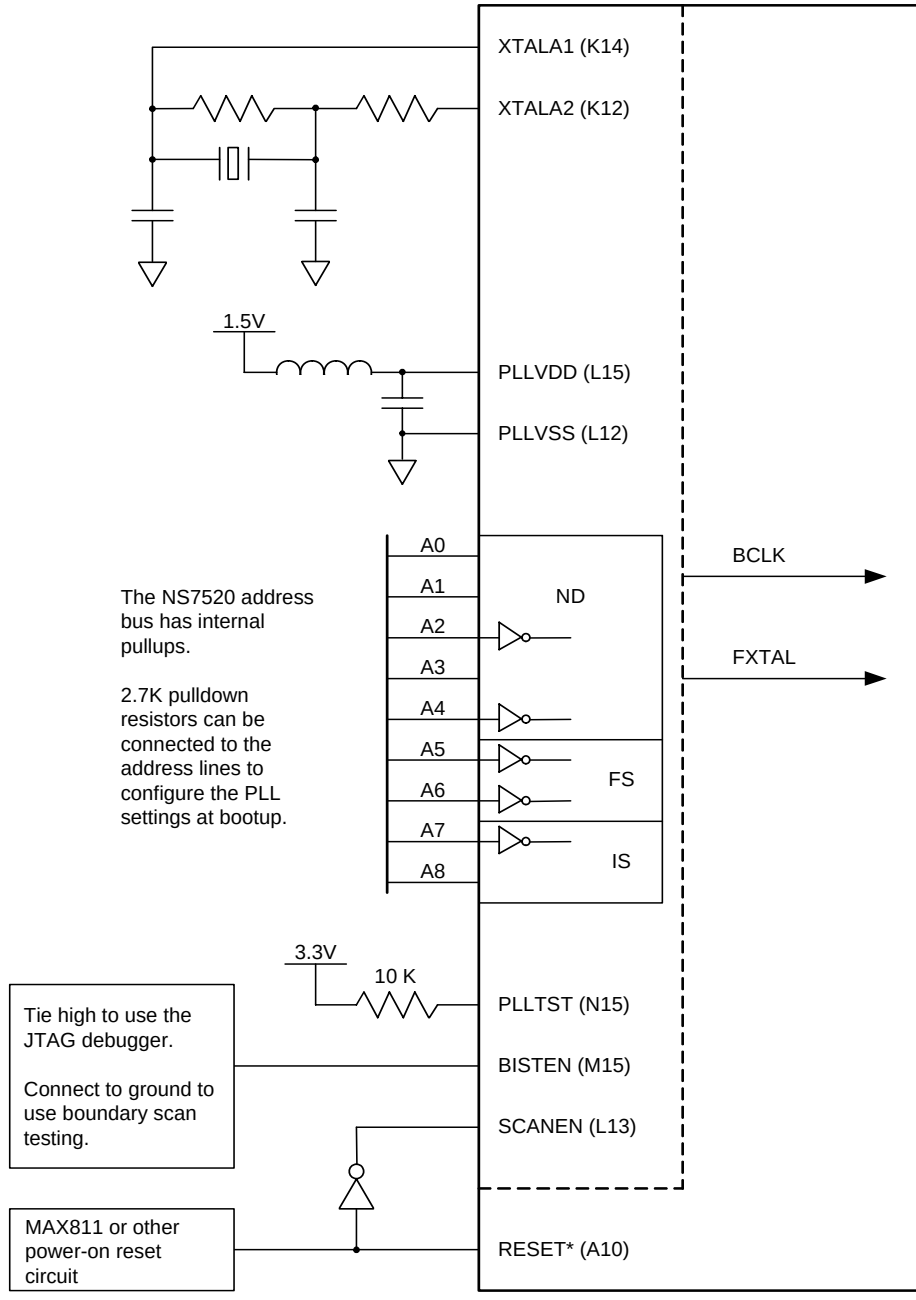


Figure 5: PLL mode hardware configuration

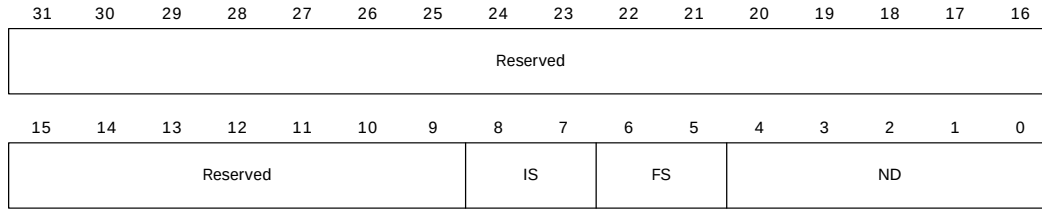
Setting the PLL frequency

Three fields — IS (charge pump current), FS (output divider), and ND (PLL multiplier) — in the PLL Settings register control the behavior of the PLL circuit. You cannot write to the PLL Settings register directly, however; it is configured in one of these ways:

1. On bootup, the PLL Settings register is configured by reading the values on address lines A[8:0]. The address lines have internal pullups. The normally high values can be changed to 0 by connecting 2.7K pulldown resistors.
2. The PLL Settings register is configured by writing to the PLL Control register. Only the ND field can be reconfigured this way.

PLL Settings register: Setting the PLL frequency on bootup

The PLL Settings register, FFB0 0040, is initialized at bootup by reading address lines A[8:0]. Only the ND field can be changed by writing a new bus speed to the PLLCNT register in the PLL Control register.



Bits	Access	Mnemonic	Reset	Description
D31:09	N/A	Reserved	N/	N/A

Table 21: PLL Settings register bit definition

Bits	Access	Mnemonic	Reset	Description
D08:07	Read only	IS	‘b10	**Charge pump current** Sets the PLL’s charge pump current. The IS field defaults to binary ‘b10 when address lines [8:7] are not pulled down on powerup. The IS value is based on the value in the ND field. (ND+1) IS 1–3 ‘b00 4–7 ‘b01 8–15 ‘b10 16–32 ‘b11
D06:05	Read only	FS	‘b00	**Output divider** Sets the PLL’s output divider. The FS field defaults to ‘b00 when address lines [6:5] are not pulled down on powerup. This is the correct setting for all frequencies and should never be adjusted.
D04:00	Read only	ND	‘b01011	**PLL multiplier** Sets the PLL’s multiplier, which determines BCLK frequency. BCLK frequency is based on tis formula: BCLK = (crystal/4) (ND + 1) The ND field defaults to ‘b01011 to produce 55MHz (with a 18.432MHz crystal) when address lines A[4:0] are not pulled down on powerup.

Table 21: PLL Settings register bit definition

The next table shows the 32 frequencies that can be produced with an 18.432MHz crystal. A 0 on an address indicates that a 2.7K pulldown resistor must be connected to that address line. The table shows the IS, FS, and ND fields, and the resulting value in the PLL Settings register.

MHz	A[8:7]	IS	A[6:5]	FS	A[4:0]	ND + 1	PLL Settings reg	Notes
4.6	01	00	11	00	10100	00001	0x00000000	
9.2	01	00	11	00	10101	00010	0x00000001	
13.8	01	00	11	00	10110	00011	0x00000002	

MHz	A[8:7]	IS	A[6:5]	FS	A[4:0]	ND + 1	PLL Settings reg	Notes
18.4	01	01	11	00	10111	00100	0x00000003	
23.0	00	01	11	00	10000	00101	0x00000084	
27.6	00	01	11	00	10001	00110	0x00000085	
32.3	00	01	11	00	10010	00111	0x00000086	
36.9	00	10	11	00	10011	01000	0x00000087	1
41.5	11	10	11	00	11100	01001	0X00000108	
46.1	11	10	11	00	11101	01010	0x00000109	2
50.7	11	10	11	00	11110	01011	0x0000010A	
55.3	11	10	11	00	11111	01100	0x0000010B	3, 4
59.9	11	10	11	00	11000	01101	0x0000010C	
64.5	11	10	11	00	11001	01110	0x0000010D	
69.1	11	10	11	00	11010	01111	0x0000010E	
73.7	11	11	11	00	11011	10000	0x0000010F	
78.3	10	11	11	00	00100	10001	0X00000190	
82.9	10	11	11	00	00101	10010	0x00000191	
87.6	10	11	11	00	00110	10011	0x00000192	
92.2	10	11	11	00	00111	10100	0x00000193	
96.8	10	11	11	00	00000	10101	0x00000194	
101.4	10	11	11	00	00001	10110	0x00000195	
106.0	10	11	11	00	00010	10111	0x00000196	
110.6	10	11	11	00	00011	11000	0x00000197	
115.2	10	11	11	00	01100	11001	0x00000198	
119.8	10	11	11	00	01101	11010	0x00000199	
124.4	10	11	11	00	01110	11011	0X0000019A	
129.0	10	11	11	00	01111	11100	0X0000019B	
133.6	10	11	11	00	01000	11101	0X0000019C	
138.2	10	11	11	00	01001	11110	0X0000019D	
142.8	10	11	11	00	01010	11111	0X0000019E	

MHz	A[8:7]	IS	A[6:5]	FS	A[4:0]	ND + 1	PLL Settings reg	Notes
147.5	10	11	11	00	01011	1000000	0X0000019F	

Notes:

1 Digi guarantees that the NS7520B-1-C36 will work at all frequencies up to 36.9MHz.

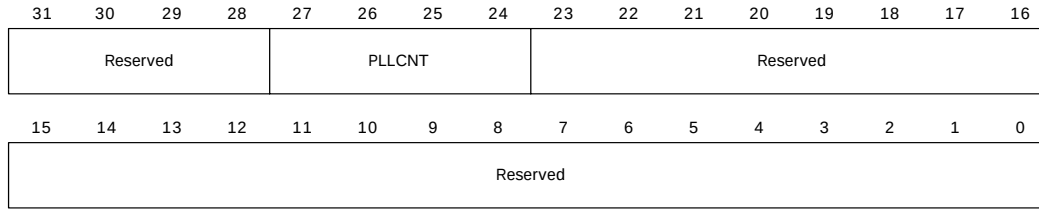
2 Digi guarantees that the NS7520B-1-C46 will work at all frequencies up to 46.1MHz.

3 Digi guarantees that the Ns7520B-1-C55 will work at all frequencies up to 55.3MHz.

4 55.3MHz is the default frequency when address lines A[8:0] are not adjusted with pulldown resistors.

PLL Control register: Setting the PLL frequency with the PLL Control register

With this method, the PLL Settings register is configured by writing to the PLL Control register, FFB0 0008.



Bits	Access	Mnemonic	Reset	Description
D31:28	N/A	Reserved	N/	N/A
D27:24	R/W	PLLCNT	0x9	**SYS_CLK frequency** Writing to this field affects the PLL frequency. See the discussion following this table.
D23:00	N/A	Reserved	N/A	N/A

Table 22: PLL Control register bit definition

The PLL frequency can be changed by writing to the PLLCNT field in the PLL Control register. At bootup, the default value in the PLLCNT field has no affect on the PLL frequency. This is because the PLL frequency set on bootup is based on the state of address lines A[8:0]. The value in the PLLCNT field must be rewritten to change the PLL frequency. The NS7520 resets whenever the PLLCNT field is changed, then starts running at the new frequency dictated by the PLLCNT value. *The readback value is not valid until software has performed a write to this register.*

When using a 18.432MHz crystal, the 4-bit value in the PLLCNT field produces frequencies of 23MHz to 92.2MHz, in increments of 4.6MHz, by changing the IS and ND fields in the PLL Settings register.

- The FS field remains the same as configured on bootup.
- The IS field is 'b10 for all standard frequencies from 36.9MHz to 55.3MHz.
- The ND field can be changed to 16 different values based on the PLLCNT. The next table shows the 16 frequencies that can be produced by changing the PLLCNT field in the PLL Control register.

MHz	PLLCNT	IS	ND + 1	PLL Settings register	Notes
23.0	0	01	00101	0x00000084	
27.6	1	01	00110	0x00000085	
32.3	2	01	00111	0x00000086	
36.9	3	10	01000	0x00000087	1
41.5	4	10	01001	0x00000108	
46.1	5	10	01010	0x00000109	2
50.7	6	10	01011	0x0000010A	
55.3	7	10	01100	0x0000010B	3
59.9	8	10	01101	0x0000010C	
64.5	9	10	01110	0x0000010D	
69.1	A	10	01111	0x0000010E	
73.7	B	11	10000	0x0000010F	
78.3	C	11	10001	0x00000110	
82.9	D	11	10010	0x00000111	

MHz	PLLCNT	IS	ND + 1	PLL Settings register	Notes
87.6	E	11	10011	0x00000112	
92.2	F	11	10100	0x00000113	

Notes:

1 Digi guarantees that the NS7520B-1-C36 will work at all frequencies up to 36.9MHz.

2 Digi guarantees that the NS7520B-1-C46 will work at all frequencies up to 46.1MHz.

3 Digi guarantees that the Ns7520B-1-C55 will work at all frequencies up to 55.3MHz.

Reset circuit sources

There are three reset circuit sources: external, watchdog, and software.

- **External reset.** Powerup reset is initiated when the RESET_ input is asserted low when power is applied. RESET_ should be driven low for at least 40ms after power has reached safe operating levels, to allow external crystals to start up. At other times, RESET_ must be one microsecond minimum.
- **Watchdog reset.** The watchdog reset is synchronous to SYS_CLK. A hardware reset condition is triggered when the signal transitions from active to inactive state.
- **Software reset.** The CPU can also trigger a software reset by writing ‘h123 and ‘h321 to the Software Service register (see "Software Service register," beginning on page 70). Unless otherwise noted, configuration registers are not reset by a software reset. When a software reset is triggered, the appropriate modules are reset for a total of 16 sys_clk periods.

NS7520 bootstrap initialization

Many internal NS7520 features are configured when the RESET pin is asserted. The address bus configures the appropriate control register bits at powerup. (See also "External oscillator mode hardware configuration," beginning on page 51.)

Address bit	Name	Description
ADDR[27]	Endian configuration	0 Little Endian configuration 1 Big Endian configuration
ADDR[26]	CPU bootstrap	0 CPU disabled; GEN_BUSER = 1 1 CPU enabled; GEN_BUSER = 0
ADDR[24:23]	CS0/MMCR[19:18] setting	00 8-bit SRAM, 63 wait-states/b00 01 32-bit SRAM, 63 wait-states/b01 10 32-bit SRAM 11 16-bit SRAM, 63 wait-states/b11
ADDR[19:09]	GEN_ID setting	GEN_ID = A[19:09], Default = `'h3ff`
ADDR[8:7]	PLL IS setting	IS = A[8:7], Default = `'b10`
ADDR[6:5]	PLL FS setting	FS = A[6:5], Default = `'b00`
ADDR[4:0]	PLL ND setting	ND = A[4:0], Default = `'b01011`

Note: The initial operating bus speed must be selected by adding pulldown to the address lines that preset the ND value (ADDR[4:0]).

GEN Module

C H A P T E R 6

The GEN module provides the NS7520 with its main system control functions, as well as the following:

- Two programmable timers with interrupt
- One programmable bus-error timer
- One programmable watchdog timer
- Two 8-bit programmable general-purpose I/O ports

Module configuration

The GEN module is configured as shown:

Address	Register
FFB0 0000	System Control register
FFB0 0004	System Status register
FFB0 000C	Software Service register
FFB0 0010	Timer 1 Control register
FFB0 0014	Timer 1 Status register
FFB0 0018	Timer 2 Control register
FFB0 001C	Timer 2 Status register
FFB0 0020	PORTA register
FFB0 0028	PORTC register
FFB0 0030	Interrupt Enable Register 1
FFB0 0034	Interrupt Enable Register 1–Set
FFB0 0038	Interrupt Enable Register 1–Clear
FFB0 0034	Interrupt Status Register 1–Enabled
FFB0 0038	Interrupt Status Register–Raw

Table 23: GEN module address configuration

GEN module hardware initialization

Many internal NS7520 configuration features are application-specific and need to be configured at powerup before the CPU begins executing bootup code. See "NS7520 bootstrap initialization" on page 60.

GEN module registers

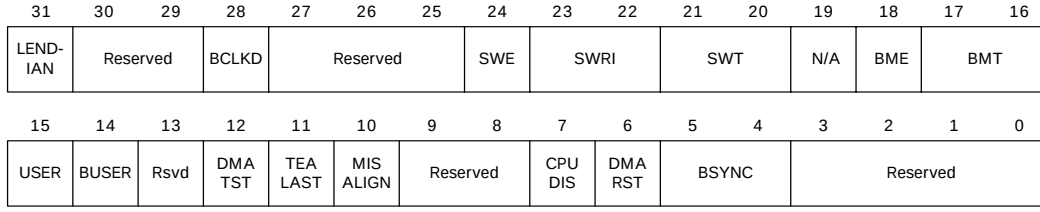
All registers are 32 bits unless otherwise noted.

System Control register

Address: FFB0 0000

General information

All bits in the System Control register are active high unless an underscore (_) appears in the signal name; the underscore indicates active low.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	LENDIAN	ADDR27	**Configure chip to run in Little Endian mode** Controls the Endian configuration for the NS7520. 1 Configures the chip to operate in Little Endian mode 0 Configures the chip for Big Endian mode
D30:29	N/A	Reserved	N/A	Initialized to and always read as 10 (full speed).

Table 24: System Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D28	R/W	BCLKD	0	**BCLK output disable** 0 BCLK output enabled 1 BCLK output forced to LOW state Shuts down the operation of the BCLK signal. Turning off the BCLK signal minimizes electro-magnetic interference (EMI) when BCLK is not required for an application.
D27:25	N/A	Reserved	N/A	N/A
D24	R/W	SWE	0	**Software watchdog enable** Set to 1 to enable the watchdog timer circuit.The watchdog timer can be configured, using SWRI, to generate an interrupt or reset condition if and when the watchdog timer expires. Once SWE is set to 1, only a hardware reset sets the bit back to 0.
D23:22	R/W	SWRI	0	**Software watchdog reset/interrupt select** Controls the action that occurs when the watchdog timer expires: 00 Software watchdog causes normal (IRQ) interrupt 01 Software watchdog causes fast (FIRQ) interrupt 10 Software watchdog causes reset 11 Reserved
D21:20	R/W	SWT	0	**Software watchdog timeout (in seconds)** Controls the timeout period for the watchdog timer. The timeout period is a function of F_{XTALE}: 00 $2^{20}/F_{XTALE}$ 01 $2^{22}/F_{XTALE}$ 10 $2^{24}/F_{XTALE}$ 11 $2^{25}/F_{XTALE}$
D19	N/A	Reserved	N/A	N/A

Table 24: System Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D18	R/W	BME	0	**Bus monitor enable** 0 Disable bus monitor operation 1 Enable bus monitor operation Required to avoid a system lockup condition that can occur when a bus master tries to address memory space that is not decoded by any peripheral. The bus monitor timer detects when a bus master is accessing a peripheral and there is no transfer acknowledge (TA_) response. When the bus monitor timer expires, the current bus cycle is terminated immediately and the current system bus master is issued a data abort indicator.
D17:16	R/W	BMT	0	Bus monitor timer Controls the timeout period for the bus monitor timer: 00 128 BCLKs (bus clocks) 01 64 BCLKS 10 32 BCLKS 11 16 BCLKS The BMT field generally is set to its maximum value, but can be set to a lower value to minimize the latency when issuing a data abort signal. The BMT field needs to be set to a value that is larger than the anticipated longest access time for all peripherals.
D15	R/W	USER	0	**Enable access to internal chip registers in CPU user mode** Controls whether applications operating in ARM user mode (rather than supervisor mode) can access internal registers within the NS7520. ■ If set to 0, and an application, operating in ARM user mode tries to access (read or write) an internal NS7520 register, the application receives a data abort. This causes a transfer in control to the data abort handler. ■ If set to 1, any application can access the NS7520 internal registers.

Table 24: System Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D14	R/W	BUSER	~ ADDR [26]	**Enable ARM CPU** Must be set to 0. When reset, BUSER defaults to the value defined by ADDR26 (see "NS7520 bootstrap initialization" on page 60).
D13	N/A	Reserved	N/A	N/A
D12	R/W	DMATST	0	**DMA module test mode** Resets the DMA controller subsystem. Also allows the ARM processor direct access to the internal context RAM found in the DMA controller. ■ When set to 1 (allow unrestricted access), the DMA controller subsystem is held in reset and the ARM processor can access all internal DMA context RAM. This is useful for diagnostic purposes. ■ When set to 0 (test mode disabled), the DMA controller subsystem operates normally. Only the bits in the DMA Control register space can be accessed by the ARM processor.
D11	R/W	TEALAST	0	**Bus interface TEA/LAST configuration** This bit can be read or written with a setting of 1 or 0, but has no effect on chip functionality.
D10	R/W	MISALIGN	0	**Bus error on misaligned cycles** 0 Disable misaligned data transfer bus abort generation 1 Generate a bus abort during a misaligned transfer When this bit is set to 1, *misaligned address transfers* cause a data abort to be issued to the offending bus master. A misaligned address transfer is defined as a half word access to an odd byte address boundary, or a full word access to either a half word or byte address boundary. This bit is useful during software debugging to detect misaligned cycles.
D09:08	N/A	Reserved	N/A	N/A

Table 24: System Control register bit definition

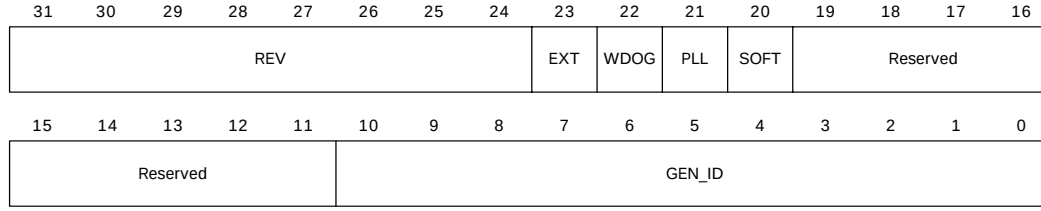
Bits	Access	Mnemonic	Reset	Description
D07	R/W	CPUDIS	-ADDR26	**CPU disable** 0 CPU operational 1 CPU reset Provides a mechanism to read back the bootstrap value of ADDR26 (see "NS7520 bootstrap initialization" on page 60). If this bit is set to 1, the CPU is disabled.
D06	R/W	DMARST	0	**DMA module reset** 0 DMA module operational 1 DMA module (held in) reset Provides a mechanism to issue a soft reset to the DMA module without affecting any other modules.
D05:04	R/W	BSYNC	0	**TA_ input synchronizer** Defines the level of synchronization performed within the NS7520 for TA_ input: 00 1-stage synchronizer 01 1-stage synchronizer 10 2-stage synchronizer 11 Do not use this setting The NS7520 can process the TA_ input signal using a 1-stage flip-flop synchronizer or a 2-stage synchronizer. A 1- or 2-stage synchronizer must be used when TA_ input is asynchronous to the BCLK signal. **Note:** The 2-stage synchronizer is preferable, as it introduces one additional BCLK of latency in the access cycle.
D03:00	N/A	Reserved	N/A	N/A

Table 24: System Control register bit definition

System Status register

Address: FFB0 0004

All bits in the System Status register, except EXT, WDOG, PLL, and SOFT, are loaded during a hardware reset only. EXT, WDOG, PLL, and SOFT are loaded during any reset. The GEN_ID value is not affected when an external jumper is changed, unless a hardware reset is executed first.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:24	R/O	REV	'h29	**NS7520 revision ID** Provides hardware identification of the NS7520 and its revision. Current NS7520 device and revision ID is: REV field: 'h29
D23	R/C	EXT	N/A	**Last reset caused by external reset** When set to 1, indicates that the RESET_ pin triggered the last hardware reset condition. This reset initializes internal parameters as described in "NS7520 bootstrap initialization" on page 60. EXT is set/reset during every reset condition. Clear this bit by writing 'hF in bits 23:20.

Table 25: System Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D22	R/C	WDOG	N/A	**Last reset caused by watchdog timer** When set to 1, indicates that a watchdog timeout occurred and generated an internal hardware reset condition. **Note:** Because the RESET_ pin is not asserted, this reset does *not* initialize internal parameters as described in "NS7520 bootstrap initialization" on page 60. WDOG is set/reset during every reset condition. Clear this bit by writing 'hF in bits 23:20.
D21	R/C	PLL	N/A	**Last reset caused by PLL update** When set to 1, indicates that the PLL was updated and required an internal hardware reset. **Note:** When the software modifies the PLL settings the RESET_ pin is not asserted, this reset does *not* initialize internal parameters as described in "NS7520 bootstrap initialization" on page 60. PLL is set/reset during every reset condition. Clear this bit by writing 'hF in bits 23:20.
D20	R/C	SOFT	N/A	**Last reset caused by software reset** When set to 1, indicates that a soft reset was triggered by software (see "Software Service register" on page 70). **Note:** Because the RESET_ pin is not asserted, this reset does *not* initialize internal parameters as described in "NS7520 bootstrap initialization" on page 60. SOFT is set/reset during every reset condition. Clear this bit by writing 'hF in bits 23:20.
D19:11	N/A	Reserved	N/A	N/A
D10:00	R/O	GEN_ID	ADDR19:09	**Product ID defined by external resistor jumpers** Defaults to the value defined by ADDR19:09 (see "NS7520 bootstrap initialization" on page 60) during a hardware reset.

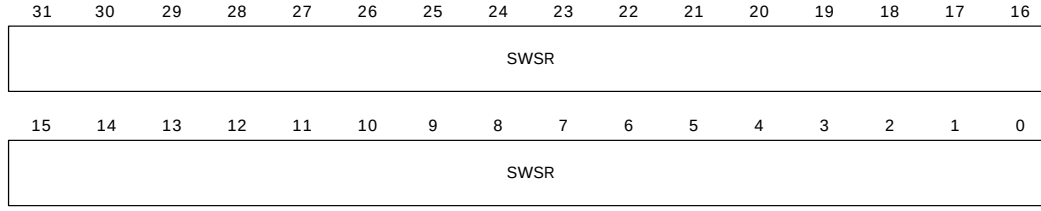
Table 25: System Status register bit definition

Software Service register

Address: FFB0 000C

The Software Service register (SWSR) acknowledges the system watchdog timer. To do so, firmware must write 'h5A and 'hA5 to the register using two separate write operations. There is no restriction on the time between the two operations, but the operations must occur in the proper sequence with the proper data values.

The Software Service register can request a software reset of the NS7520 hardware. Firmware must write 'h123 and 'h321 to the register using two separate write operations. There is no restriction on the time between the two operations, and the two operations must occur in the proper sequence with the proper data values. The processor must be in supervisor mode for the second operation.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	W	SWSR	0	**Software Service register**

Table 26: Software Service register bit definition

Timer Control registers

Address: FFB0 0010 / FFB0 0018

Timers 1 and 2 provide the CPU with programmable interval timer(s). The timers use the F_{XTALE} timing reference and an optional 9-bit prescaler or the system clock. Each timer provides a 27-bit programmable-down counter mechanism.

The CPU loads an initial count register (ITC) to define the timeout period. When the current counter decrements to zero, the counter is reloaded. The reloading of the

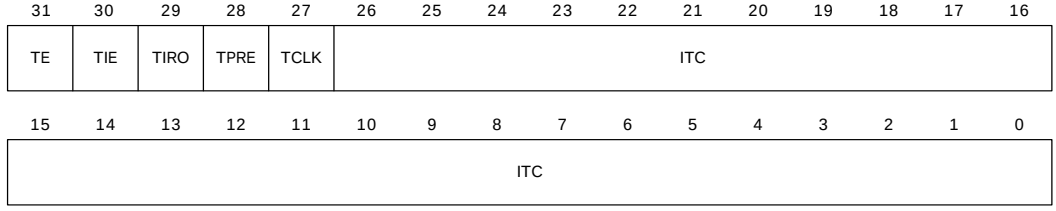
timer can be programmed to generate an interrupt to the CPU. The CPU can read the current count value (CTC) at any time.

These equations determine the timeout interval:

$TIMEOUT = [8 * (TC + 1)] / F_{XTALE}$ TCLK = 0; TPRE = 0

$TIMEOUT = [4096 * (TC + 1)] / F_{XTALE}$ TCLK = 0; TPRE = 1

$TIMEOUT = (TC + 1) / F_{SYSCLK}$ TCLK = 1; TPRE = x



Register bit definition

Bits	Access	Mnemonic	Reset	Description
D31	R/W	TE	0	**Timer enable** 1 Allows the timer to operate. 0 Resets and disables the timer. The other fields in this register should be configured before or during the same memory cycle in which TE is set to 1.
D30	R/W	TIE	0	**Timer interrupt enable** When set to 1, allows the timer to interrupt the CPU. A timer interrupt is generated when the hardware sets the TIP bit in the Timer Status register (see "Timer Status registers" on page 73).
D29	R/W	TIRO	0	**Timer interrupt mode** 0 Normal interrupt 1 Fast interrupt Controls the type of interrupt the timer asserts to the CPU.

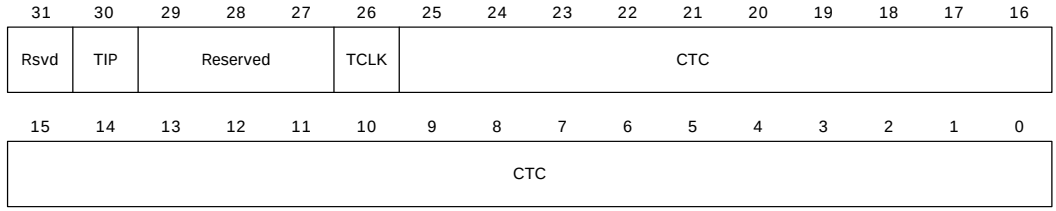
Table 27: Timer Control registers bit definition

Bits	Access	Mnemonic	Reset	Description
D28	R/W	TPRE	0	**Timer prescaler** 0 Disable 9-bit prescaler 1 Enable 9-bit prescaler Determines whether the 9-bit prescaler will be used in calculating the TIMEOUT parameter. The prescaler allows for longer TIMEOUT values. TPRE affects TIMEOUT only when F_{XTALE} is used as a time source.
D27	R/W	TCLK	0	**Timer clock source** 0 Use F_{XTALE} as timer clock source 1 Use F_{SYSCLK} as timer clock source Selects the reference clock for the timer module.
D26:00	R/W	ITC	0	**Initial timer count** Defines the TIMEOUT parameter for interrupt frequency. The TIMEOUT period is a function of the $F_{XTALE.}$

Table 27: Timer Control registers bit definition

Timer Status registers

Address: FFB0 0014 / FFB0 001C



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	N/A	Reserved	N/A	N/A
D30	R/C	TIP	0	**Timer interrupt pending** Set to 1 when the timer is enabled and the CTC value counts down to 0. TIP generates an interrupt to the CPU if the TIE bit in the Timer Control register is set. Writing a 1 to the same bit position in the Timer Status register clears the TIP bit. **Note:** TIP is set immediately when the TE bit (in the Timer Control register) is changed from 0 to 1. An interrupt occurs immediately after TE transitions from 0 to 1. If this initial interrupt causes a problem in any specific application, the software must be designed to ignore the first interrupt after TE transitions from 0 to 1.
D29:27	N/A	Reserved	N/A	N/A
D26:00	R	CTC	O	**Current timer count** Each time the CTC field reaches zero, the TIP bit is set and the CTC is reloaded with the value defined in the ITC field. The CTC continues to count back down to zero.

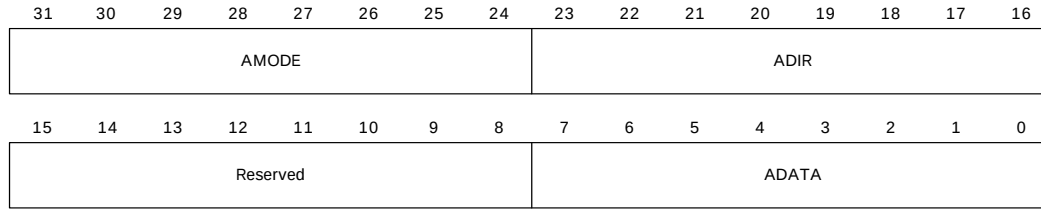
Table 28: Timer Status registers bit definition

PORTA Configuration register

Address: FFB0 0020

The PORTA register configures the PORTA general-purpose input/output (GPIO) pins.

Each of the PORTA GPIO pins can be individually programmed — as general-purpose input or output, or special function input or output — as applicable. Table 29 describes the PORTA register; Table 30 shows the different configurations for each bit.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:24	R/W	AMODE	0	**PORTA mode configuration** 0 Selects GPIO mode 1 Selects special function mode Configures the individual PORTA pins. Each bit in the AMODE field corresponds to one of the PORTA bits; D31 controls PORTA7, D30 controls PORTA6, and so on.
D23:16	R/W	ADIR	0	**PORTA data direction** 0 Selects input mode 1 Selects output mode Configures the individual PORTA pins. Each bit in the ADIR field corresponds to one of the PORTA bits; D23 controls PORTA7, D22 controls PORTA6, and so on.
D15:08	N/A	Reserved	N/A	N/A

Table 29: PORTA register bit definition

Bits	Access	Mnemonic	Reset	Description
D07:00	R/W	ADATA	0	**PORTA data register** Used when a PORTA bit is configured to operate in GPIO mode. ■ Reading the ADATA field provides the current state of the GPIO signal, regardless of its configuration mode. ■ Writing the ADATA field defines the current state of the GPIO signal when the signal is defined to operate in GPIO output mode. ■ Writing the ADATA field when configured in GPIO input mode or special function mode has no effect. Each bit in the ADATA field corresponds to one of the PORTA bits; D07 controls PORTA7, D06 controls PORTA6, and so on.

Table 29: PORTA register bit definition

PORTA Configuration

The ADIR and AMODE bits together provide independent configuration of each pin. Each column in this table denotes one of the possible configurations for each bit. If there is no entry in one of the columns (for example, `PORTA7>AMODE=1>ADIR=0`), the bit cannot be used in that configuration.

PORTA bit	AMODE = 0		AMODE = 1	
	ADIR = 0	ADIR = 1	ADIR = 0	ADIR = 1
PORTA7	GPIO IN	GPIO OUT		SER1_TXD
PORTA6	GPIO IN	GPIO OUT	DREQ1_IN	SER1_DTR_
PORTA5	GPIO IN	GPIO OUT		SER1_RTS_
PORTA4	GPIO IN	GPIO OUT	SER1_RI_/SER1_RXC IN/SER1_SPI_S_CLK IN	SER1_SPI_M_CLK OUT/SER1_OUT1/ SER1_RXC OUT
PORTA3	GPIO IN	GPIO OUT	SER1_RXD	DACK1_OUT
PORTA2	GPIO IN	GPIO OUT	SER1_DSR_	AMUX

Table 30: PORTA configuration

PORTA bit	AMODE = 0		AMODE = 1	
	ADIR = 0	ADIR = 1	ADIR = 0	ADIR = 1
PORTA1	GPIO IN	GPIO OUT	SER1_CTS	DONE1_OUT_
PORTA0	GPIO IN	GPIO OUT	SER1_SPI_S_ENABLE_ IN/SER1_DCD_/ DONE1_IN_/ SER1_TXC IN	SER1_SPI_M_ENABLE_ OUT/SER1_OUT2

Table 30: PORTA configuration

Inputs
An input that provides one or more input signals to different blocks in the chip is always connected to those blocks. The target block must be configured to use the input; similarly, those blocks that should not use the input must be configured not to use it.

Outputs
Configuring a pin for an output function also enables the tri-state driver for that pin. There should be no external driver on a pin configured for output.

READBACK
When reading the ADATA field, the data read depends on how the pin is configured:

- **Configured as GPIO output.** Reads data from the register whose data drives the pin. This can, for example, mask a short circuit on the output pin.
- **All other configurations.** Reads the state of the pin.

PORTA4
When PORTA4 is configured with `AMODE[4]=1:ADIR[4]=1`, the configuration of Serial Channel A determines the function `SER1_SPI_M_CLK_OUT, SER1_OUT1, SER1_RXC_OUT.`

PORTA2
The memory module configures the AMUX output signal. The AMUX configuration overrides the AMODE, ADIR, and ADATA fields.

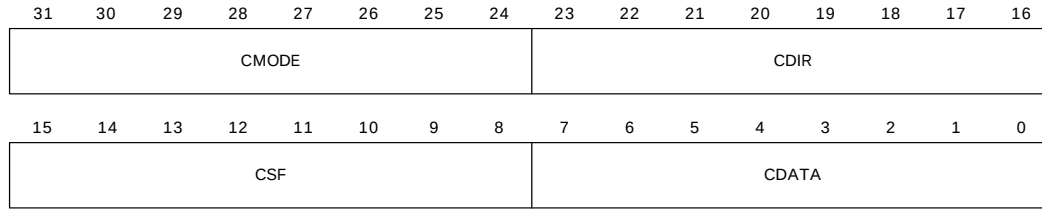
PORTA0
When PORTA0 is configured with `AMODE[0]=1:ADIR[0]=1`, the configuration of Serial Channel A determines the function `SER1_SPI_M_ENABLE_A_ OUT` or `SER1_OUT2`.

PORTC Configuration register

Address: FFB0 0028

The PORTC register configures the PORTC general-purpose input/output (GPIO pins).

Each of the PORTC GPIO pins can be individually programmed — as general-purpose input or output, or special function input or output — as applicable. Table 31 describes the PORTC register; Table 32 shows the different configurations for each bit.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:24	R/W	CMODE	‘h10	**PORTC mode** Configures the individual PORTC pins. Each bit in the CMODE field corresponds to one of the PORTC bits; D31 controls PORTC7, D30 controls PORTC6, and so on.
D23:16	R/W	CDIR	‘h10	**PORTC data direction** Configures the individual PORTC pins. Each bit in the CDIR field corresponds to one of the PORTC bits; D23 controls PORTC7, D22 controls PORTC6, and so on.
D15:08	R/W	CSF	0	**PORTC special function** Configures the individual PORTC pins. Each bit in the CSF field correspond to one of the PORTC bits. D15 controls PORTC7, D14 controls PORTC6, and so on.

Table 31: PORTC register bit definition

Bits	Access	Mnemonic	Reset	Description
D07:00	R/W	CDATA	0	**PORTC data register** Used when a PORTC bit is configured to operate in GPIO mode. ■ Reading the CDATA field provides the current state of the GPIO signal, regardless of its configuration mode. ■ Writing the CDATA field defines the current state of the GPIO signal when the signal is defined to operate in GPIO output mode. ■ Writing the CDATA field when configured in GPIO input mode or special function mode has no effect. Each bit in the CDATA field corresponds to one of the PORTC bits; D07 controls PORTC7, D06 controls PORTC6, and so on.

Table 31: PORTC register bit definition

PORTC configuration

The CSF, CDIR, and CMODE bits together provide independent configuration of each pin. Each column in this table denotes one of the possible configurations for each bit. If there is no entry in one of the columns (for example, `PORTC7>CMODE=1>CDIR=0`), the bit cannot be used in that configuration.

PORTC bit	CSF = 0			
	CMODE = 0		CMODE = 1	
	CDIR = 0	CDIR = 1	CDIR = 0	CDIR = 1
PORTC7	GPIO IN	GPIO OUT		IRQOUT_
PORTC6	GPIO IN	GPIO OUT		DTR_
PORTC5	GPIO IN	GPIO OUT		RTS_
PORTC4	GPIO IN	GPIO OUT	RIB_	RESET_OUT_
PORTC3	GPIO IN	GPIO OUT	LEVELIRQ3 = CDIR3	
PORTC2	GPIO IN	GPIO OUT	LEVELIRQ2 = CDIR2	

Table 32: PORTC configuration

PORTC bit	CSF = 0			
	CMODE = 0		CMODE = 1	
	CDIR = 0	CDIR = 1	CDIR = 0	CDIR = 1
PORTC1	GPIO IN	GPIO OUT	LEVELIRQ1 = CDIR1	
PORTC0	GPIO IN	GPIO OUT	LEVELIRQ0 = CDIR0	
PORTC bit	**CSF = 1**			
	CMODE = 0		**CMODE = 1**	
	CDIR = 0	**CDIR = 1**	**CDIR = 0**	**CDIR = 1**
PORTC7				SER2_TXD
PORTC6			DREQ2_IN	SER2_DTR_
PORTC5			REJECT_	SER2_RTS_
PORTC4			SER2_SPI_S_CLK IN/ SER2_RXC IN	SER2_SPI_M_CLK OUT/SER2_TXC OUT/ SER2_OUT1
PORTC3			SER2_RXD	DACK2_OUT
PORTC2			SER2_DSR_	RPSF_
PORTC1			SER2_CTS_	DONE2_OUT_
PORTC0	DONE2_IN_		SER2_SPI_S_ENABLE_ IN/SER2_DCD_/ SER2_TXC IN	SER2_SPI_M_ENABLE _OUT/SER2_OUT2

Table 32: PORTC configuration

Inputs

An input that provides one or more input signals to different blocks in the chip is always connected to those blocks. The target block must be configured to use the input; similarly, those blocks that should not use the input must be configured not to use it.

Outputs

Configuring a pin for an output function also enables the tri-state driver for that pin. There should be no external driver on a pin configured for output.

READBACK
When reading the CDATA field, the data read depends on how the pin is configured:

- **Configured as GPIO output.** Reads data from the register whose data drives the pin. This can, for example, mask a short circuit on the output pin.
- **All other configurations.** Reads the state of the pin.

PORTC4
When PORTC is configured with `CSF[4]=1:CMODE[4]=1:CDIR[4]=1`, the configuration of Serial Channel B determines the function `SER2_SPI_M_CLK_OUT, SER2_TXC_OUT,` or `SER2_OUT1`. Following external reset, `CSF[4]=0:CMODE[4]=1:CDIR[4]=1`; that is, set to drive RESET_ output.

PORTC0
When PORTC0 is configured with `CSF[0]=1:CMODE[0]=1:CDIR[0]=1`, the configuration of Serial Channel B determines the function `SER2_SPI_M_ENABLE_OUT` or `SER2_OUT2`.

PORTC[3:0]
These pins can be programmed individually to generate level-sensitive interrupts. Level-sensitive interrupts generate an interrupt when the input signal matches the state of the corresponding DIR bit. The interrupt condition persists until the input signal changes state or the configuration is changed.

Interrupts

There are two wires that go to the CPU core and interrupt the processor:

- **IRQ.** Normal interrupt.
- **FIRQ.** Fast interrupt.

FIRQ has higher priority than IRQ, providing a simple two-tier priority scheme to the interrupt system. Most sources of interrupts on the NS7520 come from the IRQ line. FIRQ interrupt sources are the three built-in timers and the watchdog timer, controlled by the Timer 1/2 and Status registers and the System Control register, respectively.

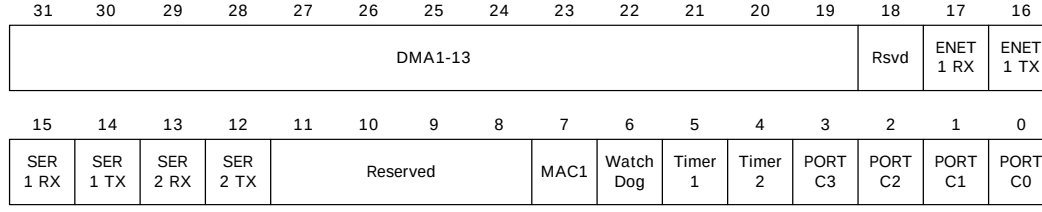
Interrupts come from different sources on the chip and are managed with Interrupt Control registers. Interrupts can be enabled/disabled on a per-source basis using the Interrupt Enable registers. These registers serve as masks for the different interrupt sources.

Interrupt controller registers

Address: FFB0 0030 / 0034 / 0038

There are five pairs of registers in the interrupt controller:

- **Interrupt Enable register.** A read/write location for reading and writing all interrupt enable bits as a typical register.
- **Interrupt Enable Set/Interrupt Status Enabled registers.** Perform two different functions depending on whether a register is read or written:
  - When read, the register indicates the current state of all enabled interrupts.
  - When written, a 1 in a bit position sets that interrupt enable; a 0 in a bit position has no effect.
- **Interrupt Enable Clear/Interrupt Status Raw registers.** Perform two different functions depending on whether a register is read or written:
  - When read, the register indicates the current state of all interrupts regardless of the state of the enables.
  - When written, a 1 in a bit position clears that interrupt enable; a 0 in a bit position has no effect.



Register bit assignment

All registers use the same 32-bit layout.

Bits	Access	Mnemonic	Reset	Description
D31:19	R/W	DMA1–13	0	The DMA1 through DMA13 bit positions correspond to interrupts sourced by DMA channel 1 through 13.
D18	N/A	Reserved	N/A	N/A
D17	R/W	ENET1RX	0	The ENET1RX bit position corresponds to an interrupt sourced by the Ethernet receiver.
D16	R/W	ENET1TX	0	The ENET1TX bit position corresponds to an interrupt sourced by the Ethernet transmitter.
D15	R/W	SER 1 RX	0	The SER 1 RX bit position corresponds to an interrupt sourced by the Serial Channel A receiver.
D14	R/W	SER 1 TX	0	The SER 1 TX bit position corresponds to an interrupt sourced by the Serial Channel A transmitter.
D13	R/W	SER 2 RX	0	The SER 2 RX bit position corresponds to an interrupt sourced by the Serial Channel B receiver.
D12	R/W	SER 2 TX	0	The SER 2 TX bit position corresponds to an interrupt sourced by the Serial Channel B transmitter.
D11:08	N/A	Reserved	N/A	N/A
D07	R/W	MAC1	0	The MAC1 bit position corresponds to an interrupt sourced by the Ethernet MAC 1.
D06	R/W	WATCHDOG	0	The WATCHDOG bit position corresponds to an interrupt condition sourced by the watchdog timer.
D05	R/W	TIMER 1	0	The TIMER 1 bit position corresponds to an interrupt condition sourced by the TIMER 1 module.
D04	R/W	TIMER 2	0	The TIMER 2 bit position corresponds to an interrupt condition sourced by the TIMER 2 module.
D03	R/W	PORTC3	0	The PORTC3 bit position corresponds to an interrupt condition sourced by the PORTC3 input.
D02	R/W	PORTC2	0	The PORTC2 bit position corresponds to an interrupt condition sourced by the PORTC2 input.

Table 33: Interrupt Enable registers bit definition

Bits	Access	Mnemonic	Reset	Description
D01	R/W	PORTC1	0	The PORTC1 bit position corresponds to an interrupt condition sourced by the PORTC1 input.
D00	R/W	PORTC0	0	The PORTC0 bit position corresponds to an interrupt condition sourced by the PORTC0 input.

Table 33: Interrupt Enable registers bit definition

Memory Controller Module

CHAPTER 7

The memory (MEM) module provides a glueless interface to external memory devices such as flash, DRAM, and EEPROM. The memory controller contains an integrated DRAM controller, and supports five unique chip select configurations.

About the MEM module

The MEM module monitors the BBus interface for access to the BUS module; that is, any access not addressing internal resources. If the BBus for the access corresponds to a Base Address register in the MEM module, the module provides the memory access signals and responds to the BBus with the appropriate completion signal.

The MEM module can be configured to interface with FP, EDO, or synchronous DRAM (SDRAM), although the NS7520 cannot interface with more than one device type at a time.

MEM module hardware initialization

Many NS7520 configuration features are application-specific and need to be configured at powerup before the CPU boots. See "NS7520 bootstrap initialization" on page 60.

PORTA2, the DRAM address multiplexer, provides for an external address mux for SDRAM, FP DRAM, or EDO DRAM.

Pin configuration

The NS7520 uses several pins to support SRAM and DRAM devices. The MEM module controls the following signals: ADDR[27:0], CS[4:0], CAS[3:0], WE_ and OE_. Table 34 shows how MEM module pins are configured for different memory types.

Mode	A27:14	A13:0	CS*x*	CAS3_	CAS2_	CAS1_	CAS0_	OE_	WE_
SRAM	Address	Address	CS[4:0]_	———	———	———	———	OE_	WE_
DRAM-FP	Address	Internal mux	RAS_	CAS3_	CAS2_	CAS1_	CAS0_	OE_	WE_
DRAM-EDO	Address	Internal mux	RAS_	CAS3_	CAS2_	CAS1_	CAS0_	OE_	WE_

Table 34: MEM module pin configuration by memory type

Mode	A27:14	A13:0	CS*x*	CAS3_	CAS2_	CAS1_	CAS0_	OE_	WE_
SDRAM	Address	Internal mux	CS[4:0]_	RAS_	CAS_	WE_	A10/AP	——	——

Table 34: MEM module pin configuration by memory type

- **Chip select configured for SRAM.** The MEM module controls the CS[4:0]_, OE_, and WE_ signals. The address from the current bus master is driven directly to A[27:0].
- **Chip select configured for FP or EDO DRAM.** The MEM module can be programmed to drive a multiplexed address on A[13:0], and drives the remainder of the address from the current bus master to A[27:14].
  - The CS[4:0]_ signals provide the RAS_ function.
  - The CAS_ signals provide the CAS_ function.
  - The OE_ and WE_ signals provide the output and write enables, respectively.
- **Chip select configured for SDRAM.** The MEM module can be programmed to drive a multiplexed address on A[13:0], and drives the remainder of the address from the current bus master to A[27:14].
  - The CS[4:0]_ signals provide the CS[4:0]_ function.
  - The CAS3_ signal provides the RAS_ function.
  - The CAS2 function provides the CAS_ function.
  - The CAS1_ signal provides the WE_ function.
  - The CAS0_ signal provides the A10/AP multiplexed signal. The A10/AP multiplexes between the A10 pin for the DRAM and the auto precharge indicator. The CAS0_ signal must always be connected to the SDRAM A10 pin.
- SDRAMs require the *DQM* function. The BE[3:0]_ signals provides the DQM function.

MEM module configuration

The MEM module is configured as shown in Table 35. Each chip select contains an identical set of three registers that appear on a boundary of 'h10 bytes.

Each register is 32 bits unless otherwise noted.

Address	Mnemonic	Register
FFC0 0000	MMCR	Memory Module Configuration register
FFC0 0010	BAR0	Chip Select 0 Base Address register
FFC0 0014	OR0A	Chip Select 0 Option Register A
FFC0 0018	OR0B	Chip Select 0 Option Register B
FFC0 0020	BAR1	Chip Select 1 Base Address register
FFC0 0024	OR1A	Chip Select 1 Option Register A
FFC0 0028	OR1B	Chip Select 1 Option Register B
FFC0 0030	BAR2	Chip Select 2 Base Address register
FFC0 0034	OR2A	Chip Select 2 Option Register A
FFC0 0038	OR2B	Chip Select 2 Option Register B
FFC0 0040	BAR3	Chip Select 3 Base Address register
FFC0 0044	OR3A	Chip Select 3 Option Register A
FFC0 0048	OR3B	Chip Select 3 Option Register B
FFC0 0050	BAR4	Chip Select 4 Base Address register
FFC0 0054	OR4A	Chip Select 4 Option Register A
FFC0 0058	OR4B	Chip Select 4 Option Register B

Table 35: Memory controller register map

Setting the chip select address range

Each chip select should be configured to respond to a different portion of the memory map. Do this by setting the appropriate fields in the Chip Select Base Address and Chip Select Option registers.

The BASE field in the Chip Select Base Address register defines the starting address of the chip select address space. The MASK field identifies those address bits, from A[31:12], that are used in the address decoding function.

- A 1 in the MASK field indicates that the associated address bit is to be used in the decoding process.
- A 0 in the MASK field indicates that the associated address is to be ignored in the address decoding process.

Note: When accessing a static memory device, the maximum value of the base address is `0x03000000`.

To determine whether a BBus address is associated with a peripheral chip select, apply the following Boolean equation (chip select decode is TRUE when):

```
(({MASK,000} & 32-bit logical address) = = ({BASE,000} & {MASK,000}))
```

- `{MASK,000}` is the MASK field concatenated with three nibbles of 0.
- `{BASE,000}` is the BASE field concatenated with three nibbles of 0.
- `&` is the logical AND function.
- `==` is the *equal to* operator.

The NS7520 supports only 28 address bits. Each chip select is limited to 256 Mbytes.

Memory Space

The MASK field can alias a memory location in different areas. The memory space associated with an individual chip select does not have to be continuous. Using the MASK field, you can set gaps in a single chip select range, essentially assigning the same chip select to different portions of the memory map.

Example

A 16 Mbyte device can be addressed in four different 64 Mbyte address locations by using a MASK value of `'hF3000`. The peripheral is addressable at these address locations:

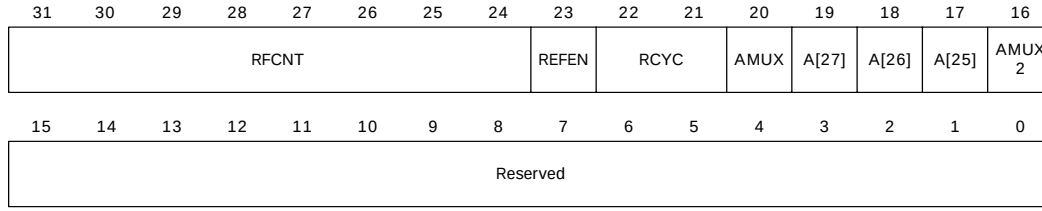
- `BASE + 'h00000000`
- `BASE + 'h04000000`
- `BASE + 'h08000000`
- `BASE + 'h0C000000`

Memory Module Configuration register

Address: FFC0 0000

The Memory Module Configuration register (MMCR) defines basic MEM module configurations.

Note: The software reset command issued by the GEN module Software Service register has no effect on any MEM Module Configuration registers.



Bits	Access	Mnemonic	Reset	Description
D31:24	R/W	RFCNT	0	**Refresh count value** Refresh period = $[(2*RFCNT + 2) * 2] / F_{XTALE}$ Defines the refresh period for the memory controller when FP/EDO DRAM or SDRAM is being used. All DRAMs require a periodic refresh cycle. You determine the specific cycle. **Note:** There is a small performance and power penalty for programming a refresh period smaller than required. The memory controller generates CAS before RAS refresh cycles for all DRAM types.
D23	R/W	REFEN	0	**Enable DRAM refresh** 0 Disable DRAM refresh 1 Enable DRAM refresh Must be set to 1 when DRAMs are used.

Table 36: MMCR bit definition

Bits	Access	Mnemonic	Reset	Description
D22:21	R/W	RCYC	0	**Refresh cycle count** The refresh timing, based on the sample timing diagram shown in "fp_refresh_cycles" on page 294, is selected by the value in this field for all chip selects programmed for FP/ EDO DRAM operation. This field has no effect on SDRAM.
D20	R/W	AMUX	0	**Enable external address multiplexing** 0 Disable external address multiplexing on PORTA2 for all DRAM banks 1 Enable external address multiplexing on PORTA2 for all DRAM banks Controls whether the NS7520 uses its internal DRAM address multiplexer. When set to 0, the NS7520 uses the internal DRAM address multiplexer for all DRAM access cycles. The DRAM RAS/CAS multiplexed address is routed through the A13:A0 pins. See "NS7520 DRAM address multiplexing" on page 105 for more information. When set to 1, the NS7520 uses an external DRAM multiplexer. The RAS/CAS address select signal is routed out the PORTA2 signal. The external DRAM RAS/CAS multiplexer uses the PORTA2 signals to determine when to switch the address multiplexer
D19	R/W	A[27]	0	**Enable A27 output** 0 CS0OE_ is driven out the A27 pin 1 The A27 signal is driven out the A27 pin The bit settings determine how the NS7520 uses this signal. See "A27 and A26 bit settings" on page 92 for more information.

Table 36: MMCR bit definition

Bits	Access	Mnemonic	Reset	Description
D18	R/W	A[26]	0	**Enable A26 output** 0 CS0WE_ is driven out the A26 pin 1 The A26 signal is driven out the A26 pin. The bit settings determine how the NS7520 uses this signal. See "A27 and A26 bit settings" on page 92 for more information.
D17	R/W	A25	1	**Enable A25 output** Always set to 0; used for address bit 25.
D16	R/W	AMUX2	0	**Internal/External/RAS/CAS mux** 0 Normal operation 1 Drive the DRAM MUX control out PORTA2, regardless of the AMUX and DMUXS settings Used to drive the DRAM RAS/CAS address multiplexing control signal out the PORTA2 pin, regardless of the AMUX setting. When set to 1, the memory controller drives the DRAM RAS/CAS address multiplexing control signal out the PORTA2 pin, for an external address multiplexer to use for DRAM RAS/CAS address multiplexing control.
D15:00	N/A	Reserved	N/A	N/A

Table 36: MMCR bit definition

A27 and A26 bit settings

The A27 bit setting determines how the A27 signal is used by the NS7520. CS0OE_ is generated by an internal logical AND of the CS0_ and OE_ signals. The CS0OE_ signal goes active low when both CS0_ and OE_ are active low.

The A26 bit setting determines how the A26 signal is used by the NS7520. The CS0WE_ signal is generated by an internal logical AND of the CS0_ and WE_ signals. the CS0WE_ signal goes active low when both CS0_ and WE_ are active low.

When enabled, these signals maximize the read access timing for external memory peripherals attached to CS0. When using CS0OE_, the CS0 peripheral's chip select input is attached to GND, and the read-access time for that peripheral is referenced

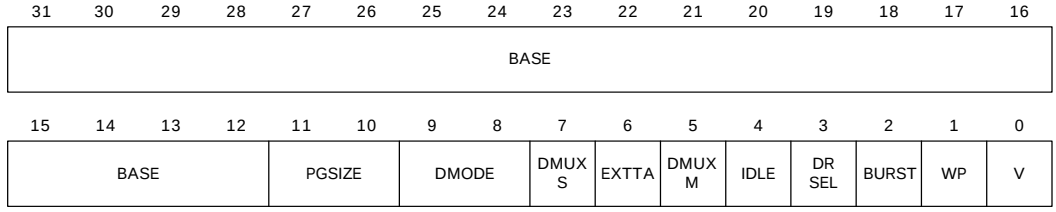
from address instead of chip select. The NS7520 provides the address signals during the earliest part of each memory cycle. The CS0OE_ and CS0WE_ signals are connected to the OE_ and WE_ input for the CS0 peripheral.

Chip Select Base Address register

Address: FFC0 0010/20/30/40/50

The Chip Select Base Address register defines the base starting address for the chip select.

Note: The V bit is set to 1 on hardware reset for chip select 0 only.



Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	BASE	0	**Base address** Determines the physical base address of the memory peripheral chip select. This 20-bit field represents the 20 most significant bits of the physical address. To derive the BASE field from a 32-bit physical address, remove the three least significant digits; for example, for a physical address of `'h00200000`, the BASE field is set to `'h00200`. When accessing a static memory device, the maximum value of the base address is `0x03000000`. See "Setting the chip select address range" on page 88 for more information.

Table 37: Chip Select Base Address register bit definition

Bits	Access	Mnemonic	Reset	Description
D11:10	R/W	PGSIZE	0	**Peripheral page size** Defines the page size for the attached peripheral with this equation: $2^{(6-PGSIZE)}$ The NS7520 halts a burst at the address boundary defined by the PGSIZE field. This field *must* be set to `'b00` for 32-bit operation of the chip select.
D09:08	R/W	DMODE	0	**DRAM configuration mode** 00 FP DRAM 01 EDO DRAM 10 SDRAM 11 Reserved Controls the DRAM type when the memory device is configured to operate in DRAM mode. This field is used only when the DRSEL bit is set to 1 (DRAM mode). All DRAM memory peripherals must be configured as the same type of DRAM.
D07	R/W	DMUXS	0	**DRAM address multiplexer select** 0 Use internal DRAM multiplexer 1 Use external DRAM multiplexer Controls whether the NS7520 uses the internal address multiplexer for this DRAM memory peripheral. 1 indicates that an external DRAM multiplexer is to be used for this chip select, where PORTA2 determines when the external multiplexer switches the address bits. When the PORTA2 signal is active high, the external DRAM RAS/CAS address multiplexer function must drive the CAS address to the DRAM devices. The AMUX or AMUX bits in the MMCR serve as a global control when set. When either of these bits is set to 1, all DRAM peripheral devices use the external address multiplexer and the DMUXS bit is ignored.

Table 37: Chip Select Base Address register bit definition

Bits	Access	Mnemonic	Reset	Description
D06	R/W	EXTTA	0	**External TA_ configuration** 0 Generate internally 1 Generate externally Defines whether TA_ cycle termination indication is generated internally or externally. **Note:** EXTTA is not allowed for DRAM.
D05	R/W	DMUXM	0	**DRAM internal address multiplexer mode** 0 10 CAS 1 8 CAS Controls which DRAM address multiplexing style is used for this DRAM memory peripheral. See "NS7520 DRAM address multiplexing" on page 105 for more information.
D04	R/W	IDLE	0	**Force BCLK at end of memory cycle** Responds to read and write, but has no effect on memory controller operation.
D03	R/W	DRSEL	0	**DRAM select** Configures the memory peripheral to operate in DRAM mode. WE CTRL and OE CTRL bits are ignored in DRAM mode; they are used only when DRSEL is set to 0.
D02	R/W	BURST	0	**Burst memory cycle enable** Controls whether the memory peripheral device supports bursting. ■ When set to 0, burst cycles are not allowed. All memory cycles are single cycles. ■ When set to 1, burst cycles are allowed.

Table 37: Chip Select Base Address register bit definition

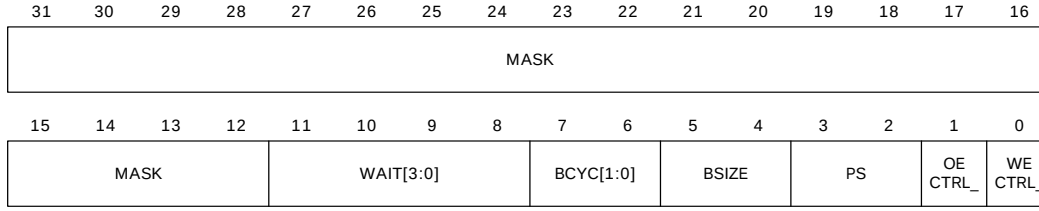
Bits	Access	Mnemonic	Reset	Description
D01	R/W	WP	0	**Write-protect the chip select** Prevents any bus master from writing to the memory device. When set to 1, all memory write-cycles are terminated immediately with a data-abort indicator. The WP bit can protect non-volatile memory devices such as flash and EEPROM.
D00	R/W	V	0	**Valid bit** Enables the chip select. When set to 1, the memory controller uses the fields in the Chip Select Base Address and Chip Select Option registers to control the behavior of the peripheral memory cycles. **Note:** It is important that you set the V bit last, after all other bits in the MMCR, Chip Select Base Address, and both Chip Select Option registers are set.

Table 37: Chip Select Base Address register bit definition

Chip Select Option Register A

Address: FFC0 0014/24/34/44/54

The Chip Select Option Register A defines the physical size of the chip select, as well as other features. Each chip select can be configured in size from 4 Kbytes to 4 Gbytes.



Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	MASK	0	**Mask Address** Controls the size of the memory peripheral decode space. Can also be used to “alias” the peripheral device in different areas of the memory map. See "Setting the chip select address range" on page 88 for more information.
D11:08	R/W	WAIT[3:0]	0 for CS[4:1], ‘b1111 for CS0	**Memory timing control fields** The WAIT field controls the number of wait states for all single cycle memory transfers and the first memory cycle of a burst transaction. The complete WAIT field is the concatenation of the four WAIT bits in this register and the two WAIT bits in Chip Select Option Register B. The BCYC field controls the number of clock cycles for the secondary portion of a burst cycle. The complete BCYC field is the concatenation of the two BCYC bits in this register and the two BCYC bits in the Chip Select Option Register B.
D07:06	R/W	BCYC[1:0]	0	

Table 38: Chip Select Option Register A bit definition

Bits	Access	Mnemonic	Reset	Description
WAIT[3:0]/BCYC[1:0] *continued*				For OE- or WE-controlled cycles, an additional BCLK cycle is added to each memory cycle. **When DRSEL = 0** ■ CS[4:0]_ is asserted for WAIT + 2 BCLK cycles in a single access. The first memory cycle of a burst access follows the timing of a single access. ■ CS[4:0]_ is asserted BCYC + 1 BCLK cycles for all cycles that follow the initial burst. If BCYC is set to 0, the controller behaves as if BCYC is set to 1. **When DRSEL = 1 and DMODE = 2'b00** ■ RAS_ is always asserted for one BCLK cycle. CAS_ is asserted for WAIT + .5 BCLK cycles in a single access. CAS_ is negated for one clock cycle between assertions.If WAIT is set to 0, the controller behaves as if WAIT is set to 1. ■ The first memory cycle of a burst access follows the timing of a single access. CAS_ is asserted BCYC + 1 BCLK cycles for all cycles that follow the initial cycle in a burst. If BCYC is set to 0, the controller behaves as if BCYC is set to 1. **When DRSEL = 1 and DMODE = 2'b01 at full speed** ■ RAS_ is always asserted for one BCLK cycle. CAS_ is asserted for WAIT BCLK cycles in a single access. CAS_ is bigoted for one clock cycle between assertions. If WAIT is set to 0, the controller behaves as if WAIT is set to 1.

Table 38: Chip Select Option Register A bit definition

Bits	Access	Mnemonic	Reset	Description
WAIT[3:0]/BCYC[1:0] *continued*				■ The first memory cycle of a burst access follows the timing of a single access. CAS_ is asserted BCYC BCLK cycles for all cycles that follow the initial cycle in a burst. If BCYC is set to 0, the controller cannot execute burst cycles. **When DRSEL = 1 and DMODE = 10** See "SDRAM," beginning on page 111.
D05:04	R/W	BSIZE	0	**Burst access size in beats** 00 2 system bus cycles in burst access 01 4 system bus cycles in burst access 10 8 system bus cycles in burst access 11 16 system bus cycles in burst access Controls the maximum number of memory cycles that can occur in a burst cycle. This field determines only the maximum number of allowable bus cycles; the current bus master can choose to burst a smaller amount. If the current bus master continues to burst, the peripheral terminates the burst when the number of memory cycles reaches the maximum allowed by this field.
D03:02	R/W	PS	0 for CS[4:1], per boot-strap for CS0	**Port size** 00 32-bit port size 01 16-bit port size 10 8-bit port size 11 Reserved Controls the size of the memory peripheral device: 8-, 16-, or 32-bits. The initial state of PS for CS0 depends on hardware initialization settings. See "NS7520 bootstrap initialization" on page 60 for more information.

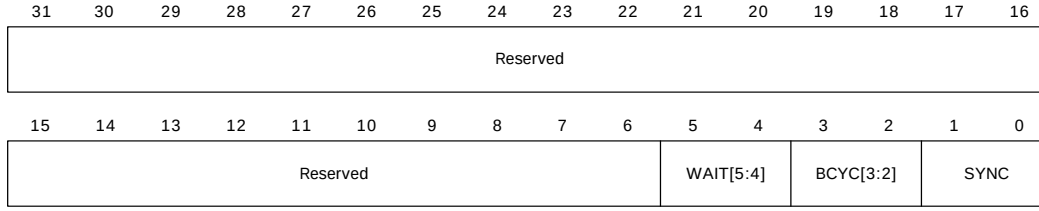
Table 38: Chip Select Option Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D01	R/W	OE CTRL_	0	**Read cycle mode** 0 Operate in *OE controlled* mode; the memory peripheral operates in a mode in which the OE_ signal is *asserted after*, and *negated while*, CS[4:0]_ is asserted. 1 Operate in *CS controlled* mode; the memory peripheral operates in a mode in which the OE_ signal is *asserted before*, and *negated after*, CS[4:0]_ is asserted Controls the access timing of the OE_ and CS[4:0]_ signals for non-DRAM memory peripherals. This bit is used only when the DRSEL bit is set to 0. When set to 1, during burst operation, there are no transitions on CS[4:0] or OE_ between single transfers with a burst.
D00	R/W	WE CTRL_	0	**Write cycle mode** 0 Operate in *WE controlled* mode; the memory peripheral operates in a mode in which the WE_ signal is *asserted after*, and *negated while*, CS[4:0]_ is asserted. 1 Operate in *CS controlled* mode; the memory peripheral operates in a mode in which the WE_ signal is *asserted before*, and *negated after*, CS[4:0]_ is asserted. Controls the access timing of the WE_ and CS[4:0]_ signals for non-DRAM memory peripherals. This bit is used only when the DRSEL bit is set to 0. When set to 1, during burst operation, there are no transitions on CS[4:0]_ or WE_ between single transfers with a burst.

Table 38: Chip Select Option Register A bit definition

Chip Select Option Register B

Address: FFC0 0018/28/38/48/58



Bits	Access	Mnemonic	Reset	Description
D31:06	N/A	Reserved	N/A	N/A
D05:04	R/W	WAIT[5:4]	0 for CS[4:1], 'b11 for CS0	**Memory timing control fields** The WAIT field controls the number of wait states for all single cycle memory transfers and the first memory cycle of a burst transaction. The complete WAIT field is the concatenation of the four WAIT bits in Chip Select Option Register A and the two WAIT bits in this register. The BCYC field controls the number of clock cycles for the secondary portion of a burst cycle. The complete BCYC field is the concatenation of the two BCYC bits in Chip Select Option Register A and the two BCYC bits in this register. For OE- or WE-controlled cycles, an additional BCLK cycle is added to each memory cycle. For information about related DRSEL and DMODE settings, see the Chip Select Option Register A bit definition table (on page 97).
D03:02	R/W	BCYC[3:2]	0	

Table 39: Chip Select Option Register B bit definition

Bits	Access	Mnemonic	Reset	Description
D01:00	R/W	SYNC	0	**TA_ input synchronizer** 00 Reserved 01 1-stage synchronizer 10 2-stage synchronizer 11 Reserved Defines the level of synchronization performed within the NS7520 for TA_ input. Used only when the chip select is configured for external TA_ mode. The NS7520 can process the TA_ input signal using a 1-stage flip-flop synchronizer, a 2-stage flip-flop synchronizer, or no synchronizer. The 1- or 2-stage synchronizers must be used if the TA_ input is synchronous to the BCLK signal. (The 2-stage synchronizer is recommended as it introduces one additional BCLK of latency in the access cycle.

Table 39: Chip Select Option Register B bit definition

Static memory (SRAM) controller

Each chip select can be configured to operate using a static memory interface. The SRAM controller supports these features:

- Synchronous mode: Transactions use the rising edge of BCLK
- Asynchronous mode: Force OE_ and WE_ pulses to be inside the active low portion of CS[4:0]_
- Burst cycle
- Programmable wait states
- Programmable base address and chip select size

Single cycle read/write

Synchronous SRAM cycles are used primarily for peripherals that can use BCLK or CS[4:0]_ as the data transfer signal. Figure 6 shows synchronous SRAM cycles.

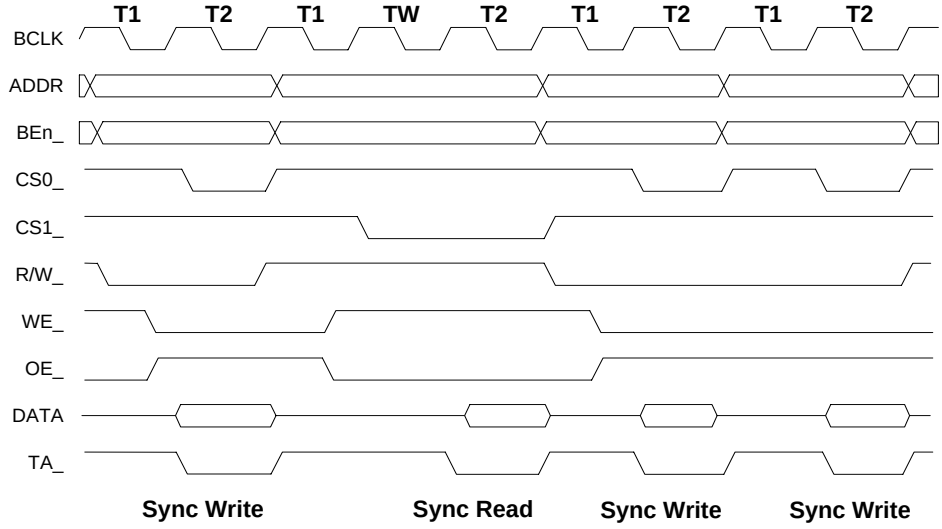


Figure 6: Synchronous SRAM cycles

- All outputs change state relative to the rising edge of BCLK with the exception of OE_ and WE_, which transition on the falling edge of BCLK.
- The OE_ and WE_ signals change state on the falling edge before CS[4:0]_ is asserted.
- OE_ and WE_ remain active until the falling edge after CS[4:0]_ is deasserted.
- The rising edge of BCLK where TA_ is low defines the end of the memory cycle (referred to as the T2 state). During synchronous read cycles, read data is sampled on the rising edge of BCLK where TA_ is low.

Asynchronous SRAM cycles guarantee that the WE_ and OE_ pulses are inside the active low pulse of CS[4:0]_. Asynchronous SRAM cycles operate a minimum of 1 wait state at all times. Figure 7 shows asynchronous SRAM cycles.

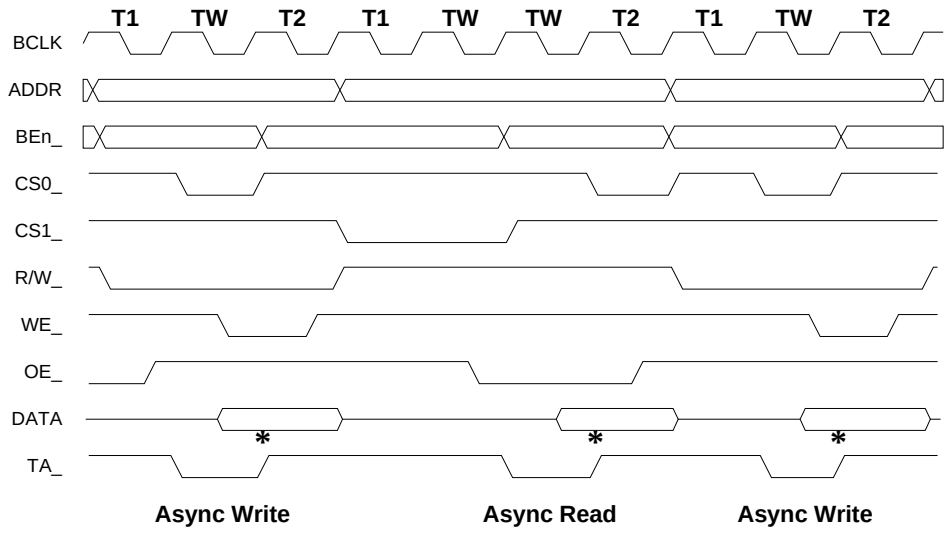


Figure 7: Asynchronous SRAM cycles

- The BE_, OE_, and WE_ signals transition based on the falling edge of BCLK.
  - The BE_, OE_, or WE_ signal transitions low on the first falling edge after CS[4:0]_ is asserted.
  - The BE_, OE_, or WE_ signal transitions high on the first falling edge after TA_ is recognized (TA_ is sampled using the rising edge of BCLK).
- The rising edge of BCLK where TA_ is low defines the last TW cycle. Read data is sampled and write data is valid on the rising edge of BCLK where TA_ is low.

Burst cycles

The SRAM controller supports both read and write burst cycles. Figure 8 shows a synchronous SRAM burst read cycle.

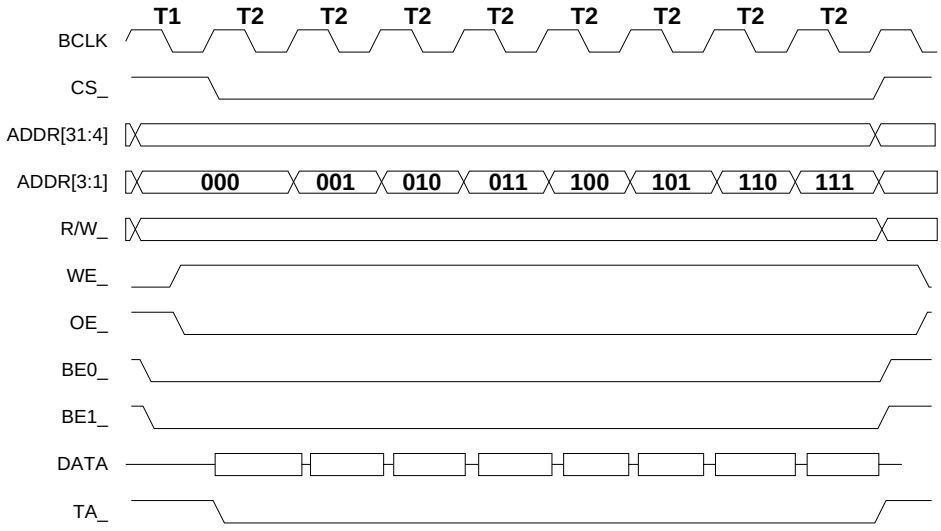


Figure 8: SRAM synchronous burst read cycle

NS7520 DRAM address multiplexing

The NS7520 can be configured to use an internal DRAM address multiplexer or an external address multiplexer. A combination of the AMUX and AMUX2 bits in the MMCR and the DMUXS bit in the Chip Select Base Address register determines which multiplexer is used.

Using the internal multiplexer

When configured to use the internal address multiplexer, the DRAM address signals are provided on system bus address pins A13:A0.

- A 32-bit DRAM peripheral connects to A13 through A2.
- A 16-bit DRAM peripheral connects to A13 through A1.
- An 8-bit DRAM peripheral connects to A13 through A0.

When a particular DRAM has less than 14 address bits, use the lower order NS7520 address bits and leave the upper NS7520 address bits disconnected. The SDRAM bank select pins must always be connected to the upper order NS7520 address pins.

Note: Never connect a bank select pin to one of the multiplexed address pins (A[13:0]).

The NS7520 supports two modes of internal address multiplexing: MODE 0 and MODE 1. The internal address multiplexing mode is configured using the DMUXM bit in the Chip Select Base Address register. Each chip select can be configured for a different address multiplexing mode.

Note: SDRAM requires MODE 1 multiplexing.

The next two tables show how the NS7520 multiplexes the logical address signal through the physical address signals during RAS and CAS timeframes. Table 40 applies to Mode 0; Table 41 applies to Mode 1.

- The top row of the table identifies the physical address connection on the NS7520 devices.
- The *DRAM* row identifies the physical address connection used on the DRAM device.
- The *RAS* row identifies the "logical" address driven by the NS7520 during the RAS portion of the RAS/CAS address multiplexing sequence.
- The *CAS* row identifies the "logical" address driven by the NS7520 during the CAS portion of the RAS/CAS multiplexing sequence.

Note that during the CAS portion of mux mode 1, the A13, A12, and A11 signals are always driven to 0. These signals must never be connected to the SDRAM bank select pins. The SDRAM bank select pins must remain stable throughout the entire memory cycle. If the bank select pins are attached to any of the multiplexed address pins, the SDRAM part fails because the bank select pins are changing state in the middle of the memory cycle.

Note: When using synchronous DRAM, A10 of the SDRAM device must always connect to CAS0_ of the NS7520. See "SDRAM," beginning on page 111, for more information.

	NS7520 multiplexed address outputs													
NS7520 pin	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin			A9			A8	A7	A6	A5	A4	A3	A2	A1	A0
RAS			18			17	16	15	14	13	11	12	10	9
CAS			19			8	7	6	5	4	3	2	1	0
	8-bit DRAM peripheral (20 address bits: 10 RAS and 10 CAS)													
NS7520 pin	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin		A9			A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAS		19			18	17	16	15	14	13	12	11	10	
CAS		20			9	8	7	6	5	4	3	2	1	
	16-bit DRAM peripheral (20 address bits: 10 RAS and 10 CAS)													
NS7520 pin	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin	A9			A8	A7	A6	A5	A4	A3	A2	A1	A0		
RAS	20			19	18	17	16	15	14	13	12	11		
CAS	21			10	9	8	7	6	5	4	3	2		
	32-bit DRAM peripheral (20 address bits: 10 RAS and 10 CAS)													

Table 40: Internal DRAM multiplexing — Mode 0

	NS7520 multiplexed address outputs															
NS7520 pin	A23	A22	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin			A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAS			21	20	19	18	17	16	15	14	13	12	11	10	9	8
CAS			0	0	0	10	20	19	7	6	5	4	3	2	1	0
	8-bit DRAM peripheral (22 address bits: 14 RAS and 8 CAS)															

Table 41: Internal DRAM multiplexing — Mode 1

	NS7520 multiplexed address outputs															
NS7520 pin	A23	A22	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin		A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAS		22	21	20	19	18	17	16	15	14	13	12	11	10	9	
CAS		22	0	0	0	10	9	8	7	6	5	4	3	2	1	
	16-bit DRAM peripheral (22 address bits: 14 RAS and 8 CAS)															
NS7520 pin	A23	A22	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DRAM pin	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0		
RAS	23	22	21	20	19	18	17	16	15	14	13	12	11	10		
CAS	23	22	0	0	0	10	9	8	7	6	5	4	3	2		
	32-bit DRAM peripheral (22 address bits: 14 RAS and 8 CAS)															

Table 41: Internal DRAM multiplexing — Mode 1

Using the external multiplexer

An external address multiplexer is required when the selected SDRAM component cannot interface with the NS7520 internal multiplexer. Although an external address multiplexer is used, the NS7520 memory controller can control the basic DRAM signal protocol. The NS7520 can be configured to output the DRAM address multiplexer signal out the PORTA2 pin, by setting the AMUX or AMUX2 bit in the MMCR or by setting the DMUXS bit in the Chip Select Base Address register.

Setting the AMUX bit indicates that the internal address multiplexer must be disabled. When AMUX is set, the NS7520 drives the address bus using standard addressing without any multiplexing, the internal multiplexer is disabled, and the multiplexer indicator is driven out the PORTA2 pin.

The AMUX2 bit allows the internal bus masters to use the internal address multiplexer, and forces the PORTA2 signal to be driven.

Setting the DMUXS bit indicates that the internal address multiplexer must be disabled when the specific chip select is activated. The NS7520 drives the address bus using standard addressing without any multiplexing, but only for the specific chip select, the internal address multiplexer is disabled, and the multiplexer indicator is driven out the PORTA2 pin.

The PORTA2 signal is driven active high during the CAS addressing portion for FP and EDO DRAM, as well as during the SDRAM write command, read command, and load mode command. The NS7520 drives the SDRAM load mode command on its lower address pins. At all other times, the PORTA2 signal is driven active low.

DRAM refresh

The NS7520 MEM module executes a refresh cycle that supports Fast Page (FP), EDO and SDRAM devices. The FP and EDO devices are refreshed using the CAS-before-RAS technique; SDRAM devices re refreshed using the REFRESH command. "fp_refresh_cycles" on page 294 provides a timing diagram of DRAM refresh cycles based in the RCYC setting. This diagram illustrates the CAS-before-RAS refresh cycles for FP and EDO DRAM. The RCYC field controls the timing of CAS and RAS in the refresh cycle.

FP/EDO DRAM controller

The memory controller module contains an integrated FP/EDO DRAM controller. Each chip select can be configured to operate using DRAM. The DRAM controller supports these features:

- FP (Fast Page) mode, EDO DRAMs, and SDRAM (see "SDRAM," beginning on page 111).
- CAS before RAS refresh operation
- Programmable refresh timer
- Background refresh cycles (refresh while another non-DRAM chip select access is in progress)

- Normal and burst (FP/EDO) cycles
- Programmable wait states for normal (also first cycle ion burst access) and burst cycles
- Programmable base address and chip select size

Single cycle read/write

Figure 9 shows FP DRAM normal read and write cycles.

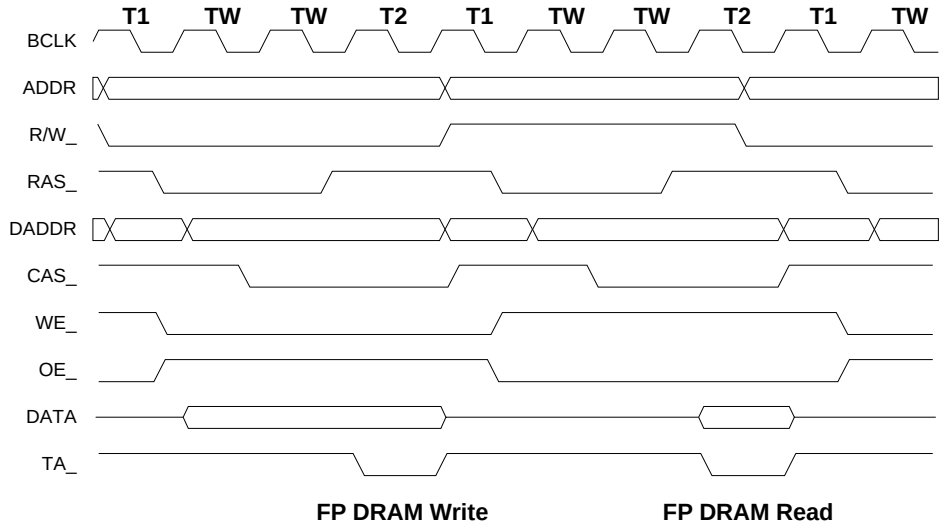


Figure 9: Normal FP DRAM bus cycles

All DRAM cycles must operate a minimum of 1 wait state. (If the controller is programmed for 0 wait states, operation is unpredictable). A single wait state DRAM cycle requires the DRAM devices to tolerate a single BCLK cycle for RAS precharge and CAS access timing.

- The CAS_ signal is deasserted on the rising edge in which TA_ is recognized.
- The RAS_ signal is deasserted on the falling edge of BCLK after CAS_ is asserted.

Important: You cannot set the PS field to `2'b11` for FP DRAM.

FP/EDO DRAM burst cycles

The DRAM controller supports both read and write burst cycles. A DRAM Burst cycle must operate with a minimum of one wait state for the first cycle and a minimum of two BCLK cycles in subsequent cycles.

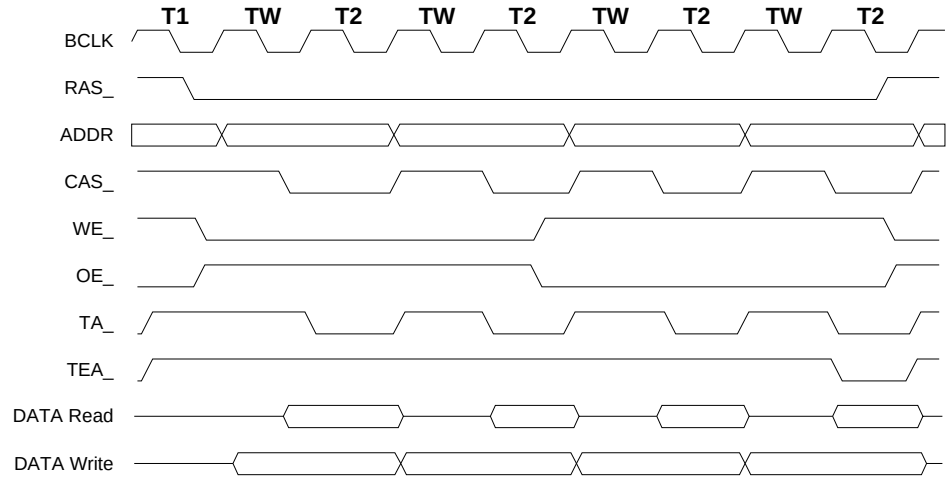


Figure 10: FP DRAM burst cycles

When bursting at full bus speed with EDO, write timing is controlled by a combination of BCYC and WAIT; otherwise, BCYC controls the length of subsequent burst cycles. (See "Chip Select Option Register A," beginning on page 97, for information about the BCYC and WAIT fields.)

SDRAM

The DMODE field in the Chip Select Base Address register configures a chip select for SDRAM (synchronous DRAM).

Important: The NS7520 cannot perform 16-bit burst operations from an x32 SDRAM.

NS7520 SDRAM interconnect

The NS7520 can interconnect to standard 16Mb and 64Mb SDRAM components, using x32, x16, and x8 SDRAM configuration.

You can use 128Mb components in the x32 configuration, but not in the x8 or x16 configurations. The NS7520 SDRAM controller does not support SDRAM with more than 8 CAS (column addressing).

x32 SDRAM configuration

Table 42 shows the interconnect between the NS7520 and SDRAM when SDRAM is used in an x32 configuration. This table shows two ways of constructing an x32 SDRAM configuration:

- Using two x16 SDARM components (referred to as *Device 1* and *Device 2*)
- Using one x32 SDRAM component

In the 2Mx32 SDRAM configuration, bank selects BA0 and BA1 are not connected in numerical order. The order is reversed so BA0 can be the same for the 4Mx32 configuration, and only BA1 needs to be swapped to upgrade.

	2x16M SDRAM signal components		**1x32 SDRAM component**	
NS7520 signal	**16 Mb (1Mx16)**	**64Mb (4Mx16)**	**64Mb (2Mx32)**	**128Mb (4Mx32)**
CS/RAS_	CS[4:0]_	CS[4:0]_	CS[4:0]_	CS[4:0]_
CAS3_	RAS_	RAS_	RAS_	RAS_
CAS2_	CAS_	CAS_	CAS_	CAS_
CAS1_	WE_	WE_	WE_	WE_
CAS0_A10/AP	A10/AP	A10/AP	A10/AP	A10/AP
BE3_	UDQM_ - Device 1	UDQM_ - Device 1	DQM3	DQM3
BE2_	LDQM_ - Device 1	LDQM_ - Device 1	DQM2	DQM2
BE1_	UDQM_ - Device 2	UDQM_ - Device 2	DQM1	DQM1
BE0_	LDQM_ - Device 2	LDQM_ - Device 2	DQM0	DQM0

Table 42: x32 SDRAM interconnect

	2x16M SDRAM signal components		1x32 SDRAM component	
NS7520 signal	**16 Mb (1Mx16)**	**64Mb (4Mx16)**	**64Mb (2Mx32)**	**128Mb (4Mx32)**
A2	A0	A0	A0	A0
A3	A1	A1	A1	A1
A4	A2	A2	A2	A2
A5	A3	A3	A3	A3
A6	A4	A4	A4	A4
A7	A5	A5	A5	A5
A8	A6	A6	A6	A6
A9	A7	A7	A7	A7
A10	A8	A8	A8	A8
A11	A9	A9	A9	A9
A12				
A13		A11		A11
A21	BA		BA1	
A22		BA0	BA0	BA0
A23		BA1		BA1
BCLK	CLK	CLK	CLK	CLK
VCC	CKE	CKE	CKE	CKE
D31-D16	D15-D00 - Device 1	D15-D00 - Device 1		
D15-D00	D15-D00 - Device 2	D15-D00 - Device 2		
D31-D00			D31-D00	D31-D00

Table 42: x32 SDRAM interconnect

x16 SDRAM configuration

Table 43 identifies the interconnect between the NS7520 and SDRAM when SDRAM is used in an x16 configuration. An x16 SDRAM configuration typically is constructed using one x16 SDRAM component.

NS7520 signal	16M SDRAM signal	64M SDRAM signal
CS/RAS_	CS[4:0]_	CS[4:0]_
CAS3_	RAS_	RAS_
CAS2_	CAS_	CAS_
CAS1_	WE_	WE_
CAS0_	A10/AP	A10/AP
BE3_	UDQM_	UDQM_
BE2_	LDQM_	LDQM_
BE1_	N/A	N/A
BE0_	N/A	N/A
A1	A0	A0
A2	A1	A1
A3	A2	A2
A4	A3	A3
A5	A4	A4
A6	A5	A5
A7	A6	A6
A8	A7	A7
A9	A8	A8
A10	A9	A9
A11		
A12		A11
A13		
A20	BA	
A21		BA0

Table 43: x16 SDRAM interconnect

NS7520 signal	16M SDRAM signal	64M SDRAM signal
A22		BA1
BCLK	CLK	CLK
VCC	CKE	CKE
D31-D16	D15-D00	D15-D00

Table 43: x16 SDRAM interconnect

x8 SDRAM configurations

Table 44 identifies the interconnect between the NS7520 and SDRAM when SDRAM is used in an x8 configuration. An x8 SDRAM configuration typically is constructed using one x8 SDRAM component.

NS7520 signal	16M SDRAM signal	64M SDRAM signal
CS/RAS_	CS[4:0]_	CS[4:0]_
CAS3_	RAS_	RAS_
CAS2_	CAS_	CAS_
CAS1_	WE_	WE_
CAS0_	A10/AP	A10/AP
BE3_	DQM_	DQM_
BE2_	N/A	N/A
BE1_	N/A	N/A
BE0_	N/A	N/A
A0	A0	A0
A1	A1	A1
A2	A2	A2

Table 44: x8 SDRAM interconnect

NS7520 signal	16M SDRAM signal	64M SDRAM signal
A3	A3	A3
A4	A4	A4
A5	A5	A5
A6	A6	A6
A7	A7	A7
A8	A8	A8
A9	A9	A9
A10		
A11		A11
A13		
A19	BA	
A20		BA0
A21		BA1
BCLK	CLK	CLK
VCC	CKE	CKE
D31-D16	D15-00	D15-00

Table 44: x8 SDRAM interconnect

SDRAM A10/AP support

The SDRAM A10/AP signal multiplexes one of the RAS signals with an auto-precharge command indicator. During the active command, the A10/AP signal must provide the DRAM A10 logical RAS address value. During the read and write commands, the AP signal provides the auto-precharge command indicator. The NS7520 drives 0 during read and write commands. During the precharge command, the A10/AP signal indicates whether all banks should be precharged. The NS7520 always causes all banks to be precharged.

The NS7520 provides the A10/AP multiplexing function using the CAS0_ pin. During the active command, the CAS0_ pin is driven with the logical value of one of the address bits A[21:18] as a function of the port size configuration defined in Chip Select Option Register A and the mux mode defined in the Chip Select Base Address register.

This table defines which address bit is driven in each situation:

Mux mode	x32	x16	x8
0	20	19	18
1	21	20	19

During the read or write commands, the NS7520 drives a 0 on the CAS0_ pin (indicating that automatic precharge should not be performed). During the precharge command, the NS7520 drives a 1 on the CAS0_ pin (indicating that all banks should be precharged.)

Command definitions

SDRAMs operate according to a series of *command codes* that are issued while chip select input is active low. The command codes are registered using the low-to-high transition of the synchronous clock. NS7520 implementation requires that all SDRAMs are synchronized to the NS7520 BCLK signal.

Command	CSx_	A13:0	CAS3_ RAS#	CAS2_ CAS#	CAS1_ WE#	CAS0_ A10/AP
Inhibit	1	X	X	X	X	X
NOP	0	X	1	1	1	X
Active	0	Bank/row	0	1	1	A10
Read	0	Column	1	0	1	0
Write	0	Column	1	0	0	0
Burst Term	0	X	1	1	0	X
Precharge	0	X	1	1	0	X
Refresh	0	X	0	0	1	X

Table 45: SDRAM command definitions

Command	CSx_	A13:0	CAS3_ RAS#	CAS2_ CAS#	CAS1_ WE#	CAS0_ A10/AP
Load mode	0	Op-code	0	0	0	0

Table 45: SDRAM command definitions

Memory timing fields — SDRAM

The WAIT configuration in the Chip Select Option register provides the SDRAM T_{RCD} and T_{RP} parameters. When WAIT is configured with a value of 0, the active and precharge commands can be followed immediately by another command on the next active edge of BCLK. When WAIT is configured with a value larger than 0, wait states are inserted after the active and precharge commands before another command can be issued.

The BCYC configuration in the Chip Select Option register provides the SDRAM CAS latency parameter. The BCYC field must be set to a value of `CAS latency - 1`. The NS7520 can support SDRAMs that have a CAS latency specification between 1 and 4 BCLK clocks, as shown:

CAS latency	BCYC configuration
1	00
2	01
3	10
4	11

BSIZE configuration

The BSIZE configuration in the Chip Select Option register provides the SDRAM burst length parameter. The BSIZE field is set as shown:

BSIZE	Burst length
00	2 words (not supported)
01	4 words (not supported)
10	8 words (not supported)

BSIZE	Burst length
11	Full page

The JEDEC SDRAM standard requires the SDRAM load mode command to set burst length to `full page` to recognize burst terminate commands. Not all SDRAMs require this setting. If you want to set burst length to a value other than `full page`, follow these steps:

1 Set BSIZE to full page.

2 Read the address within the range of the chip select.

3 Set BSIZE to the value you want to use.

The load mode command is issued as part of the first memory read cycle loading `full page` into the SDRAM Control register. The load mode command is executed only on the first memory cycle following power reset.

SDRAM Mode register

The SDRAM Mode register is loaded during the first memory cycle to the SDRAM device after the V bit is set to 1 in the Chip Select Base Address register. The MEM module loads the SDRAM Mode register with the values shown in Table 46.

Address	Field	Value
11:9	Write burst mode	001 - Single location access
8:7	Operating mode	00 - Standard operation
6:4	CAS latency	BCYC + 1
3	Burst type	0 - Sequential
2:0	Burst length	See "BSIZE configuration" on page 118.

Table 46: SDRAM Mode register settings

The load mode command is issued after at least one memory read cycle has been executed to the SDRAM device.

SDRAM read cycles

Figure 11 and Figure 12 provide timing examples for SDRAM normal and burst reads, respectively, with WAIT and BCYC configured with a value of 0.

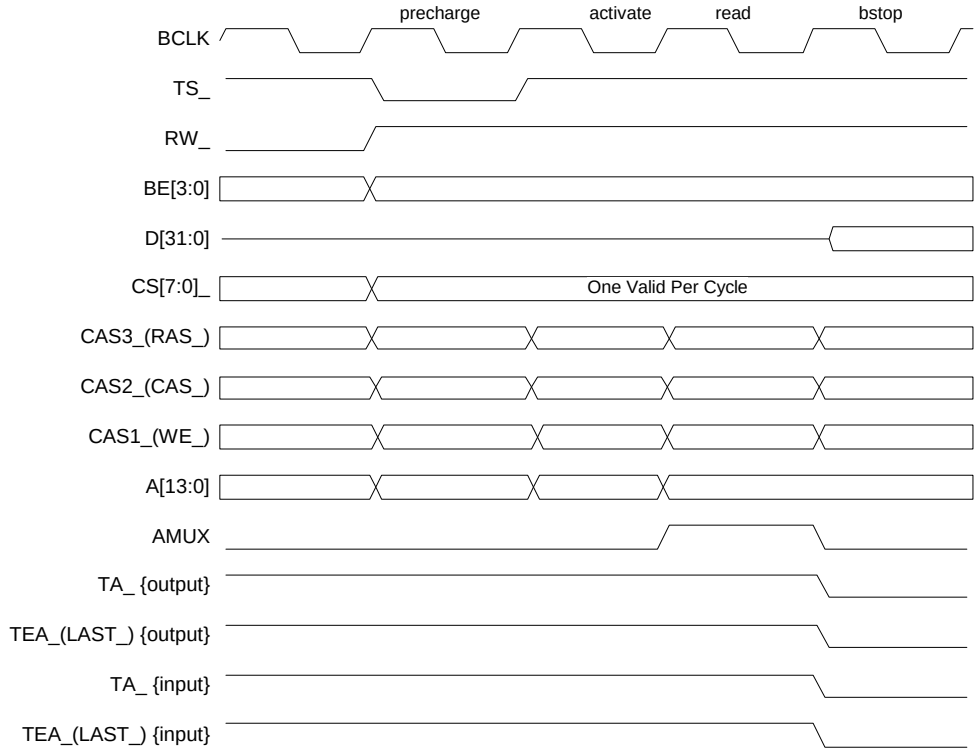


Figure 11: SDRAM normal read

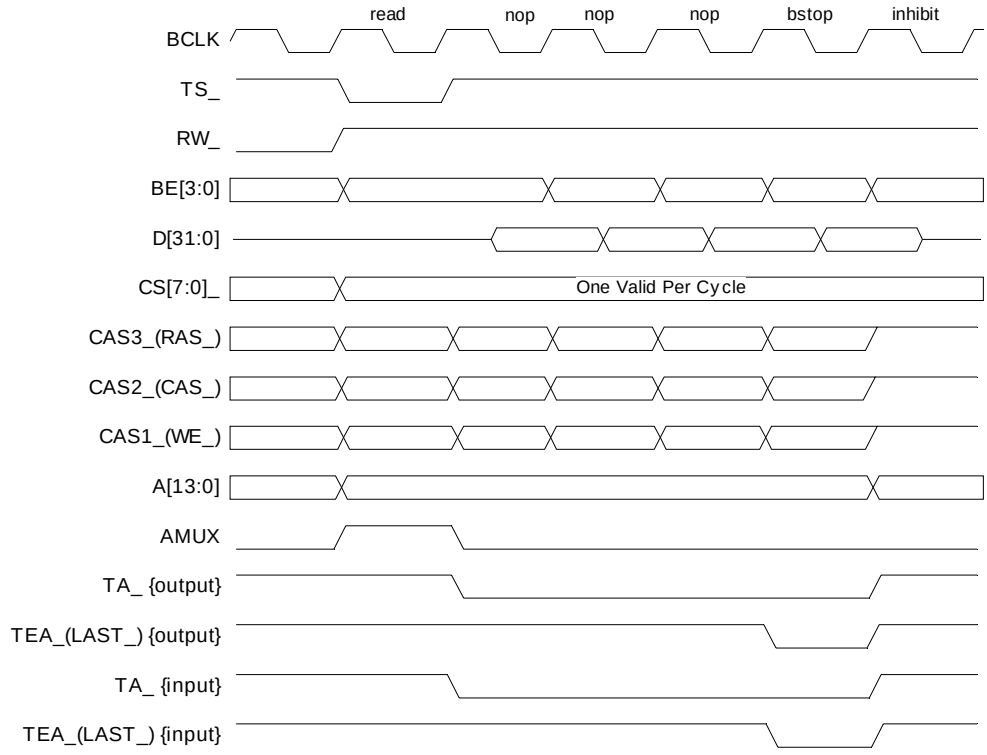


Figure 12: SDRAM burst read

- The precharge command is issued, when necessary, during the T1 phase of a normal or burst read cycle. The precharge command is issued only when the row selection for the NS7520 has changed since the last access. Each chip select maintains a 14-bit register identifying the last row accessed. Additional clock cycles are inserted between the precharge command and the active command, depending on the WAIT configuration.
- The active command is always issued after the precharge command, and selects a newly activated row address. Additional clock cycles are inserted between the active command and the read command, depending on the WAIT configuration.
- The read command is issued in either the T1 or TW states, depending on whether a precharge command was required. The read command selects the starting column address for the current burst read operation. Additional

wait states are inserted after the read command, depending on the value of the BCYC configuration. The BCYC configuration identifies the CAS latency specification for the SDRAM.

- The burst stop command is issued at the end of the current burst read operation. The SDRAM continues to burst read data for an additional number of BCLK cycles after the burst stop command is issued. The number of cycles is calculated as `CAS latency - 1`. When the CAS latency value is greater than one, additional wait states are inserted between T2 and the next system bus cycle to account for the delay. These additional bus cycles are identified as *TX* states.

SDRAM write cycles

Figure 13 and Figure 14 provide timing diagrams for SDRAM normal and burst writes, respectively, with WAIT and BCYC configured with a value of 0.

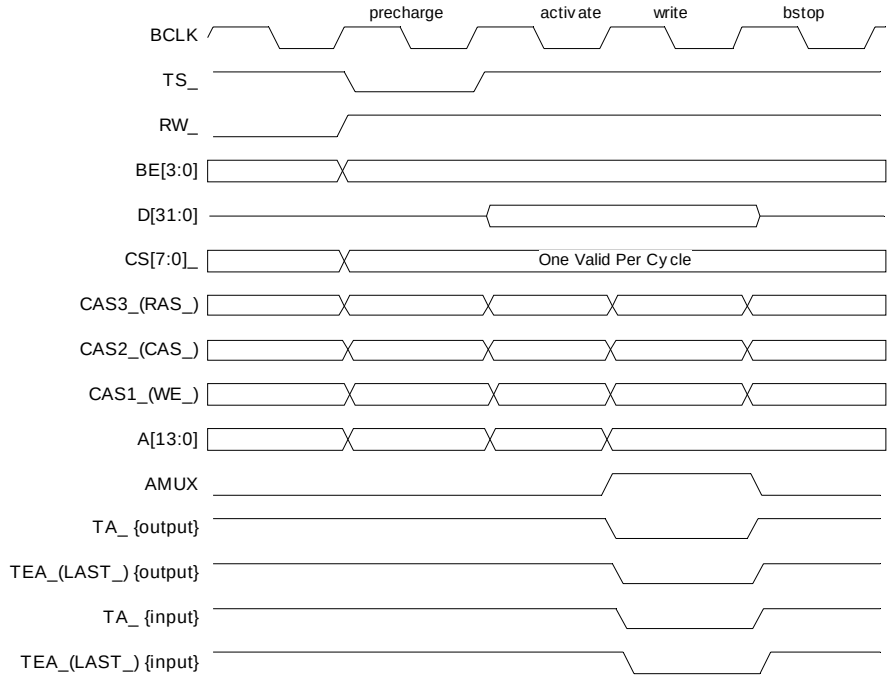


Figure 13: SDRAM normal write

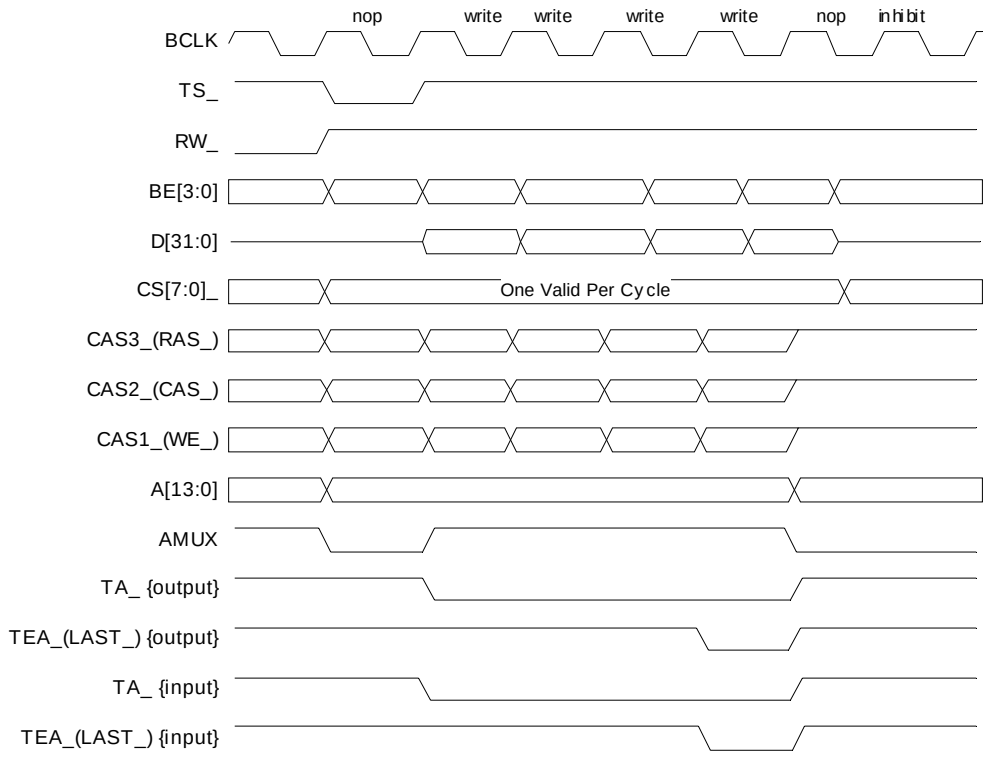


Figure 14: SDRAM burst write

- The precharge command is issued, when necessary, during the T1 phase of a normal or burst write cycle. The precharge command is issued only when the row selection for the chip select has changed since the last address. Each chip select maintains a 14-bit register identifying the last row accessed. Additional clock cycles are inserted between the precharge command and the active command, depending on the WAIT configuration.
- The active command is always issued after the precharge command. The active command selects a newly activated row address. Additional clock cycles are inserted between the active command and the read command, depending on the WAIT configuration.

- The write command is always issued during the T2 state since data is available only at that time. If the precharge and active commands are not required, a NOP is inserted in the T1 state of the write cycle.

Peripheral page burst size

Note: This section applies only to internal TA and TEA.

The peripheral device must provide a minimum burst page size of 64 bytes in order for the NS7520 to support bursting to or from a memory peripheral device.

The NS7520 can begin a burst memory access on any address boundary. The NS7520 continues the burst until one of the following conditions is met:

- The current bus master terminates the burst.
- The memory controller terminates the burst because the number of bus cycles for the burst cycle reached the maximum as defined in the BSIZE field of the Chip Select Option register.
- The current memory address has reached a 64-byte page boundary.

The bus master can decide to terminate the burst at any time. The memory controller limits the maximum number of bus cycles that can occur using the BSIZE field. The memory controller also terminates the burst when a 64-byte page boundary is encountered.

Example

- The memory peripheral is configured as a 32-bit peripheral.
- The memory peripheral has a 16-byte page size.
- The BSIZE field is set to the maximum value `11`, which implies a maximum of 16 bus cycles for each burst.
- The bus master begins a 16-byte burst cycle starting with an address offset of `'hC`.

The bus master begins a 16-byte burst cycle starting at 'hC. It expects to receive four long words of data from address offsets 'hC, 'h10, 'h14, and 'h18. The memory controller will allow the full burst because BSIZE allows a total of 16 long words to be accessed.

The memory controller does a normal access to address 'hC using the WAIT field to determine access timing. The memory controller then follows with 3 burst beats to address offsets 'h10, 'h14, and 'h18 using the BCYC field to determine the access timing.

The memory peripheral properly delivers the data at address offset 'hC. The peripheral, however, has trouble providing the value at 'h10. The NS7520 is operating with the BCYC burst timing and the memory peripheral needs an access cycle using the WAIT timing, as the access address has crossed the memory peripheral's page boundary.

If a memory peripheral has a page size less than 64 bytes, then, the memory peripheral can interface only with the NS7520, using burst cycles, when both the WAIT and BCYC fields result in the same number of clock cycles for normal and burst cycles.

You can use one of these combination:

WAIT	BCYC
0	1
1	2
2	3

DMA Module

CHAPTER 8

The NS7520 supports 13 DMA channels. Each channel moves blocks of data between memory and a memory peripheral. Each block transfer is defined by a descriptor of two words (memory-to-peripheral) or three words (memory-to-memory) in circular buffers maintained by the CPU. As each block transfer completes, the CPU need update only the descriptor for the completed block transfer, minimizing CPU intervention.

DMA module

Each DMA controller has a state machine and a block of static RAM referred to as *context RAM*.

- The context RAM contains the current state of each DMA channel.
- The single state machine supports all DMA channels in parallel, by context switching from channel to channel.

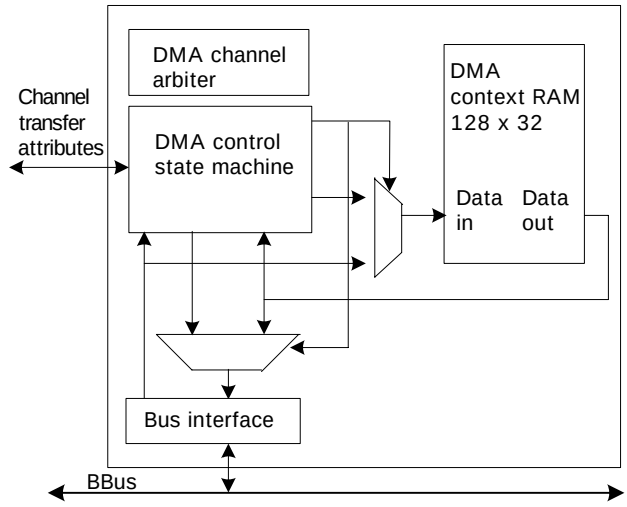


Figure 15: DMA controller block diagram

The DMA controller arbiter determines which channel the state machine currently is operating.

Fly-by operation transfers

There are two modes of fly-by operation:

- **Read mode.** Transfers data from memory to a peripheral device.
- **Write mode.** Transfers data from a peripheral device to memory.

When configured for fly-by operation, the DMA controller transfers data between one of the NS7520 internal peripherals and a memory location. The DMA controller does

not touch this data; rather, it controls the flow of data through the BBus and provides the external address for a single data transfer operation. Figure 16 provides a simple representation of DMA fly-by mode:

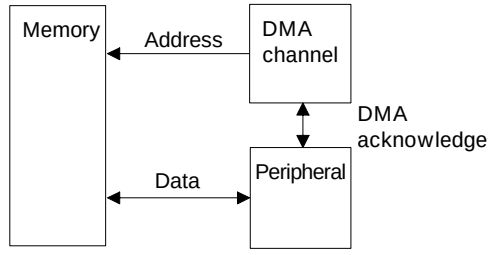


Figure 16: DMA fly-by transfers

Memory-to-memory operation

When configured for memory-to-memory (read-to-write) operations, the DMA controller uses a temporary holding register between read and write operations. Two memory cycles are executed — each read cycle is followed by a write cycle. Figure 17 provides a simple representation of DMA memory-to-memory mode:

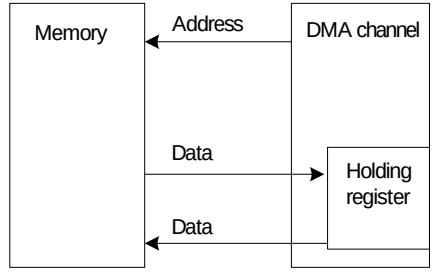


Figure 17: DMA memory-to-memory transfer

DMA buffer descriptor

All DMA channels operate using a buffer descriptor. Each DMA channel remains idle until enabled using the CE bit in the DMA Control register (see "DMA Control register," beginning on page 136). When started, a DMA channel reads the DMA buffer descriptor pointed to by the Buffer Descriptor Pointer register (see "Buffer Descriptor Pointer register," beginning on page 136). When the current descriptor is completed, the next descriptor is accessed from a circular buffer.

Each DMA buffer descriptor requires two 32-bit words for fly-by mode and three 32-bit words stored on four-word boundaries for memory-to-memory operations. Multiple buffer descriptors are located in 1024-byte circular buffers. The first buffer descriptor address is provided by the DMA channel's buffer descriptor pointer. Subsequent buffer descriptors follow the first descriptor. The final buffer descriptor is defined with its W (wrap) bit set. When the DMA channel encounters the W bit, the channel wraps around to the first descriptor. If the DMA channel does not encounter a descriptor with the W bit set, the channel wraps at the 1024-byte address boundary.

Each DMA channel can address a maximum of 128 fly-by or 64 memory-to-memory buffer descriptors. Figure 18 and Figure 19 show each descriptor type.

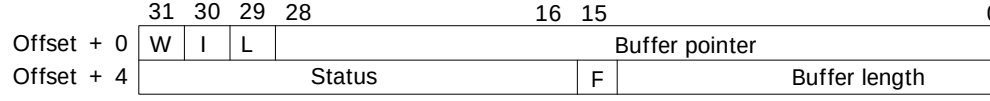


Figure 18: DMA buffer descriptor — Fly-by mode

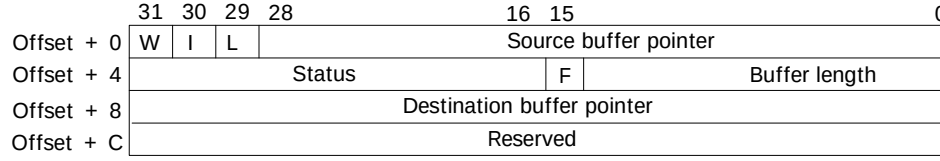


Figure 19: DMA buffer descriptor — Memory-to-memory mode

Buffer descriptor bit definitions

Bit	Description
W	Wrap bit. ■ When set (W = 1), tells the DMA controller that this is the last buffer descriptor within the continuous list of descriptors. The next buffer descriptor is found using the initial DMA channel buffer descriptor pointer. ■ When not set (W = 0), the next buffer descriptor is found using an offset from the current buffer descriptor; that is, the DMA channel continues to address buffer descriptors until it finds a descriptor with W = 1 or reaches a 1024-byte boundary.
I	When set, tells the DMA controller to issue an interrupt to the CPU when the buffer is closed because of normal channel completion. The interrupt occurs regardless of the normal completion interrupt enable configuration for the DMA channel.
L	When set, tells the DMA controller that this buffer descriptor is the last descriptor and completes an entire message frame. The DMA controller uses this bit to signal the peripheral. Use this bit when multiple descriptors are chained together to form a data frame.
F	**For fly-by operations,** when set, indicates that the buffer is full. A DMA channel sets this bit after filling a buffer and clears the bit after emptying a buffer. A DMA channel does not try to fill a buffer when the F bit is set, nor does it try to empty a buffer when the F bit is not set. When the F bit is modified by firmware, the firmware driver must write to the DMA Status/Interrupt Enable register to activate an idle DMA channel (see "DMA Status/Interrupt Enable register," beginning on page 142). If a DMA channel encounters a descriptor where F indicates it cannot move data, a maskable interrupt is generated. **For memory-to-memory operations,** the F bit must be set to 1 to begin DMA operation.

Table 47: Buffer descriptor bit definitions

Buffer descriptor field definitions

Field	Description
Source buffer pointer/Buffer pointer	The source buffer pointer field identifies the starting location of the data buffer. The source buffer pointer can start on any byte boundary for fly-by memory-to-peripheral operations. The source buffer must be aligned on 32-bit boundaries to support peripheral-to-memory operations.
Status	Peripherals can use the 16-bit status field to store transmit and receive status words. The status field is updated when the current descriptor is completed.
Buffer length: Peripheral-to-memory	In fly-by peripheral-to-memory operations, indicates the maximum number of bytes available in the receive buffer pointed to by the source buffer pointer. After filling the receive buffer with peripheral data, the DMA controller updates this field with the actual receive data byte count.
Buffer length: Memory-to-peripheral	In fly-by memory-to-peripheral operations, indicates the number of bytes to move to the peripheral device, pointed to by the source address pointer. After completing the transfer, the DMA controller updates this field in a transmit buffer descriptor with the actual data byte count (useful during error conditions).
Buffer length: Memory-to-memory	Indicates the number of bytes to move between the source and destination locations. After completing a memory-to-memory transfer, the DMA controller updates this field with the remaining data byte count (useful during error conditions).
Destination buffer pointer	Used only when the DMA channel is configured for memory-to-memory operations. The destination address pointer must start on the same byte boundary as the source address pointer. If the source and destination byte boundaries are different, the data operand size must be set for 8-bit operations (see the SIZE field in "DMA Control register," beginning on page 136).

Table 48: Buffer descriptor field definitions

DMA channel assignments

Any of the 13 channels in the DMA controller can be configured for memory-to-memory mode. Those channels assigned to a peripheral (see Table 49, "DMA channel assignments," on page 133) can be configured for fly-by mode.

- *FB write* indicates fly-by peripheral-to-memory.
- *FB read* indicates fly-by memory-to-peripheral.
- *MM* indicates memory-to-memory.
- DMA channels 3/5 and 4/6 can interface with external peripheral devices using DMA handshake signals multiplexed through the GPIO pins in PORTA and PORTC.

Channel	Base address	DMA channel peripheral	Fly-by mode
1	'hFF90 0000	Ethernet port 1 receiver	FB write
2	'hFF90 0080	Ethernet port 1 transmitter	FB read
3	'hFF90 00A0		
4	'hFF90 00C0		
5	'hFF90 00E0		
6	'hFF90 0100		
7	'hFF90 0120	SER channel 1 receiver	FB write
8	'hFF90 0140	SER channel 1 transmitter	FB read
9	'hFF90 0160	SER channel 2 receiver	FB write
10	'hFF90 0180	SER channel 2 transmitter	FB read
11	'hFF90 01A0		MM
12	'hFF90 01C0		MM
13	'hFF90 01E0		MM

Table 49: DMA channel assignments

DMA channel registers

All registers are 32-bit unless otherwise noted.

Address map

The next table sows the address map for the DMA channel configuration registers.

The DMA Control register should be written to enable the DMA channel only after all other registers and descriptors are valid.

Address	Description
FF90 0000	DMA 1 “A” Buffer Descriptor Pointer register
FF90 0010	DMA 1 “A” Control register
FF90 0014	DMA 1 “A” Status register
FF90 0020	DMA 1 “B” Buffer Descriptor Pointer register
FF90 0030	DMA 1 “B” Control register
FF90 0034	DMA 1 “B” Status register
FF90 0040	DMA 1 “C” Buffer Descriptor Pointer register
FF90 0050	DMA 1 “C” Control register
FF90 0054	DMA 1 “C” Status register
FF90 0060	DMA 1 “D” Buffer Descriptor Pointer register
FF90 0070	DMA 1 “D” Control register
FF90 0074	DMA 1 “D” Status register
FF90 0080	DMA 2 Buffer Descriptor Pointer register
FF90 0090	DMA 2 Control register
FF90 0094	DMA 2 Status register
FF90 00A0	DMA 3 Buffer Descriptor Pointer register
FF90 00B0	DMA 3 Control register
FF90 00B4	DMA 3 Status register
FF90 00C0	DMA 4 Buffer Descriptor Pointer register
FF90 00D0	DMA 4 Control register

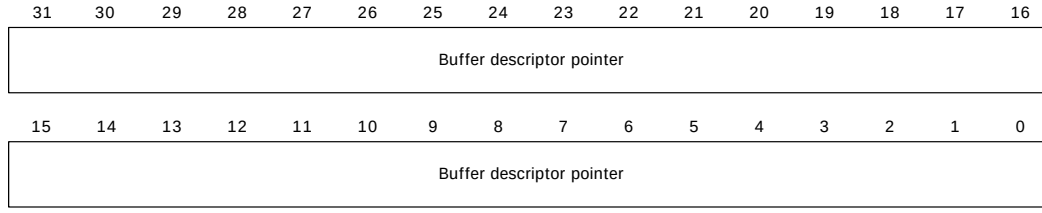
Address	Description
FF90 00D4	DMA 4 Status register
FF90 00E0	DMA 5 Buffer Descriptor Pointer register
FF90 00F0	DMA 5 Control register
FF90 00F4	DMA 5 Status register
FF90 0100	DMA 6 Buffer Descriptor Pointer register
FF90 0110	DMA 6 Control register
FF90 00114	DMA 6 Status register
FF90 0120	DMA 7 Buffer Descriptor Pointer register
FF90 0130	DMA 7 Control register
FF90 0134	DMA 7 Status register
FF90 0140	DMA 8 Buffer Descriptor Pointer register
FF90 0150	DMA 8 Control register
FF90 0154	DMA 8 Status register
FF90 0160	DMA 9 Buffer Descriptor Pointer register
FF90 0170	DMA 9 Control register
FF90 0174	DMA 9 Status register
FF90 0180	DMA 10 Buffer Descriptor Pointer register
FF90 0190	DMA 10 Control register
FF90 0194	DMA 10 Status register
FF90 01A0	DMA 11 Buffer Descriptor Pointer register
FF90 01B0	DMA 11 Control register
FF90 01B4	DMA 11 Status register
FF90 01C0	DMA 12 Buffer Descriptor Pointer register
FF90 01D0	DMA 12 Control register
FF90 01D4	DMA 12 Status register
FF90 01E0	DMA 13 Buffer Descriptor Pointer register
FF90 01F0	DMA 13 Control register
FF90 01F4	DMA 13 Status register

Buffer Descriptor Pointer register

Address: FF90 0000 / 20 / 40 / 60 / 80 / A0 / C0 / E0 / 100 / 120 / 140 / 160 / 180 / 1A0 / 1C0 / 1E0

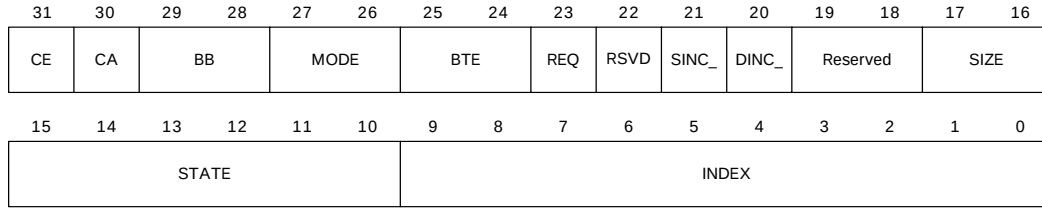
The Buffer Descriptor Pointer register contains the address of the first buffer descriptor in a contiguous list of descriptors.

DMA channel 1 is unique in that it supports four different buffer descriptors: A, B, C, and D. Each buffer descriptor contains a separate ring of descriptors, and identifies a block of data buffers with a different size. This feature allows the Ethernet receiver to choose the optimum buffer size for the incoming packet.



DMA Control register

Address: FF90 0010 / 30 / 50 / 70 / 90 / B0 / D0 / F0 / 110 / 130 / 150 / 170 / 190 / 1B0 / 1D0 / 1F0



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	CE	0	**DMA channel enable** Set *only* after the other channel mode bits are set. The DMA channel begins reading the first buffer descriptor when CE is set to 1.
D30	W	CA	0	**Channel abort request** When set, causes the current DMA operation to complete and the buffer to be closed. CA is not cleared automatically after the requested abort is complete; firmware must clear the bit after recognizing CAIP active.
D29:28	R/W	BB	0	**Bus bandwidth** Determines how often the DMA channel can arbitrate for access to the bus. 00 100% (no limit) — The DMA channel can arbitrate each time. 01 75% — The DMA channel can arbitrate 3 out of 4 times. 10 50% — The DMA channel can arbitrate 2 out of 4 times. 11 25% — The DMA channel can arbitrate 1 out of 4 times.
D27:26	R/W	MODE	0	**DMA operation mode** 00 Fly-by write (peripheral-to-memory) 01 Fly-by read (memory-to-peripheral) 10 Memory-to-memory (source-to-destination) 11 Reserved Identifies the data transfer mode. Each DMA channel can be configured to operate in either fly-by or memory-to-memory mode.

Table 50: DMA Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D25:24	R/W	BTE	0	**Burst transfer enable** Determines whether the DMA channel can use burst transfers through the bus. This configuration applies to both buffer descriptor and peripheral data access. For DMA channel 1, the BB, MODE, and BTE configuration must be the same in all four control registers (A, B, C, and D). **Fly-by mode:** 00 1 operand 01 2 operands 10 4 operands 11 Reserved In fly-by mode, the BTE field controls the maximum number of operands that the DMA controller moves each time it acquires control of the BBus. When performing DMA to an internal peripheral, the operand size is always 32 bits. When performing DMA to an external peripheral, the size field determines the operand size. The DMA controller moves information using burst cycles. If the attached memory peripheral device cannot support bursting or the peripheral terminates the burst, the maximum number of bytes defined by BTE is not reached. **Memory-to-memory mode:** 00 No burst 01 8-byte burst 10 16-byte burst 11 Reserved In memory-to-memory mode, the BTE field controls the maximum number of bytes that the DMA controller moves each time it acquires control of the BBus. The DMA controller moves information using burst cycles. If the attached source memory peripheral device cannot support bursting or the source peripheral terminates the burst, the maximum number of bytes defined by BTE is not reached.

Table 50: DMA Control register bit definition

Bits	Access	Mnemonic	Reset	Description
BTE *continued*				The DMA delivers to the destination peripheral the same number of bytes read from the source peripheral, regardless of whether the destination peripheral can support bursting. If the destination peripheral cannot support bursting, the DMA controller issues multiple bus cycles to complete the data move. When operating in memory-to-memory mode, the SIZE field determines the size of each operand moved. It is usually more efficient to use a size of 32 bits; different SIZE values are required, however, when the address alignment boundaries for the source and destination are mismatched. If the starting source and destination boundaries are not on longword boundaries, a size of 16 or 8 bits is required. If the starting source or destination boundaries are on odd-byte boundaries, a size of 8 bits is required.
D23	R/W	REQ	0	**Channel request source** 0 Internal peripheral 1 External peripheral Allows DMA channels 3/5 and 4/6 to be attached to either an internal or external peripheral. When an internal peripheral is selected, the DMA channel is hardwired to the peripheral defined in Table 49, “DMA channel assignments,” on page 133. DMA channels 3/5 interface with an external peripheral using handshaking signals multiplexed through PORTA. ■ When REQ is set to 0 in DMA channel 3 and set to 1 in DMA channel 5, DMA channel 5 is tied to the external DMA port through PORTA. ■ When REQ is set to 1 in DMA channel 3, that channel is tied to the external DMA port through PORTA, no matter the REQ bit setting in DMA channel 5.

Table 50: DMA Control register bit definition

Bits	Access	Mnemonic	Reset	Description
REQ *continued*				DMA channels 4/6 interface with an external peripheral using handshaking signals multiplexed through PORTC. ■ When REQ is set to 0 in DMA channel 4 and set to 1 in DMA channel 6, DMA channel 6 is tied to the external DMA port through PORTC. ■ When REQ is set to 1 in DMA channel 4, that channel is tied to the external DMA port through PORTC, no matter the REQ bit setting in DMA channel 6. These bits should not be set in any channels other than DMA 3, 4, 5, or 6.
D22	N/A	Reserved	N/A	N/A
D21	R/W	SINC_	0	**Source address increment** 0 Increment source address pointer 1 Do not increment source address pointer Controls whether the source address pointer is incremented after each DMA transfer. The DMA controller uses this bit in all modes when referring to a memory address. This bit is ignored when the source is a fly-by port.
D20	R/W	DINC_	0	**Destination address increment** 0 Increment destination address pointer 1 Do not increment destination address pointer Controls whether the destination address pointer is incremented after each DMA transfer. The DMA controller uses this bit in all modes when referring to a memory address. This bit is ignored when the destination is a fly-by port. For memory-to-memory operation with `DINC_=0`, the data is written to the specified destination address and all subsequent data are written are written to the specified destination address + 4.
D19:18	N/A	Reserved	N/A	N/A

Table 50: DMA Control register bit definition

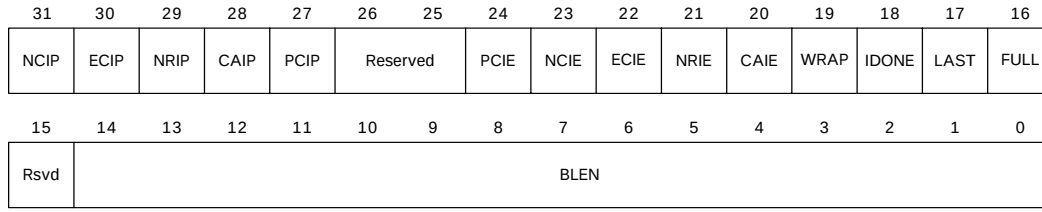
Bits	Access	Mnemonic	Reset	Description
D17:16	R/W	SIZE	0	**Data operand size** 00 32 bit 01 16 bit 10 8 bit 11 Reserved Used to define the size of each DMA transaction when the DMA controller is configured for external request mode (DMA channel 3 only) or memory-to-memory DMA mode.
D15:10	R	STATE	0	**Current DMA channel state** (shown in binary) 000000 — IDLE 000001 — Load source buffer address 000100 — Load destination buffer address 001000 — First operand 010000 — Memory-to-memory second operand 100000 — Update buffer description Describes the current state of the DMA controller state machine.
D09:00	R	INDEX	0	**Current DMA channel buffer descriptor index** Identifies the DMA controller's current byte offset pointer relative to the DMA buffer descriptor pointer.

Table 50: DMA Control register bit definition

DMA Status/Interrupt Enable register

Address: FF90 0014 / 34 / 54 / 74 / 94 / B4 / D4 / F4 / 114 / 134 / 154 / 174 / 194 / 1B4 / 1D4 / 1F4

The interrupt enable (IE) bits can be set to cause an interrupt to occur when the corresponding interrupt status bit is set in the DMA Status register. The interrupt pending (IP) bits are set to indicate begin active. These bits are cleared by writing a 1 to the same bit locations.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/C	NCIP	0	**Normal completion interrupt pending** Set when a buffer descriptor has been closed (for normal conditions) and either the NCIE bit is set or the IDONE bit is active in the current buffer descriptor. A normal DMA channel completion occurs when the BLEN count expires (gets to 0) or when a peripheral device signals completion.
D30	R/C	ECIP	0	**Error completion interrupt pending** Set when the DMA channel encounters either a bad buffer descriptor pointer or a bad data buffer pointer. When ECIP is set, the DMA channel stops until the bit is cleared by firmware; the DMA channel does not go to the next buffer descriptor. When ECIP is cleared, the buffer descriptor is tried again from where it left off. The CA bit in the DMA Control register can be used to abort the current buffer descriptor and go to the next descriptor.

Table 51: DMA Status/Interrupt Enable register bit definition

Bits	Access	Mnemonic	Reset	Description
D29	R/C	NRIP	0	**Buffer not ready interrupt pending** Set when the DMA channel encounters a buffer descriptor whose F bit is in the incorrect state. When NRIP is set, the DMA channel stops until the bit is cleared by firmware; the DMA channel does not go to the next buffer descriptor. When NRIP is cleared by firmware, the buffer descriptor is tried again.
D28	R/C	CAIP	0	**Channel abort interrupt pending** Set when the DMA channel finds that the CA bit is set in the DMA Control register. When CAIP is set, the DMA channel stops until the bit is cleared by firmware. The DMA channel automatically goes to the next buffer descriptor when CAIP is cleared. The CA bit in the DMA Control register must be cleared, using firmware, before CAIP is cleared. Otherwise, the next buffer descriptor aborts as well.
D27	R/C	PCIP	0	**Premature complete interrupt pending** Set when the DMA channel, configured to operate in fly-by *read* mode, receives an end-of-transfer indicator from the peripheral while processing a DMA buffer descriptor. The DMA channel goes to the next buffer descriptor. NCIP is set when PCIP is set, for backward compatibility.
D26:25	N/A	Reserved	N/A	N/A
D24	R/W	PCIE	0	**Premature complete interrupt enable**
D23	R/W	NCIE	0	**Normal completion interrupt enable**
D22	R/W	ECIE	0	**Error completion interrupt enable**
D21	R/W	NRIE	0	**Buffer not ready interrupt enable**

Table 51: DMA Status/Interrupt Enable register bit definition

Bits	Access	Mnemonic	Reset	Description
D20	R/W	CAIE	0	**Channel abort interrupt enable** Use these bits to enable interrupts to be generated when the associated IP bits are set. ■ In general, the NCIE bit is used for inbound (write DMA) operations. ■ The DMA buffer descriptor I bit should be used for outbound (read DMA) operations. ■ The ECIE and CAIE bits should always be enabled. ■ NCIE and the DMA buffer descriptor I bit should not be used at the same time.
D19	R	WRAP	0	**Last descriptor in descriptor list**
D18	R	IDONE	0	**Interrupt on done**
D17	R	LAST	0	**Last buffer descriptor in current data frame**
D16	R	FULL	0	**Buffer full indicator**
D15	N/A	Reserved	N/A	N/A
D[14:00]	R	BLEN	0	**Remaining byte transfer count** Use these bits for debugging purposes only. The bits serve no useful purpose to the firmware device driver.

Table 51: DMA Status/Interrupt Enable register bit definition

Ethernet transmitter considerations

When the DMA for an Ethernet transmit frame completes and the F bit is set in the buffer descriptor, the packet transmission starts immediately no matter the value in the watermark field. If the F bit is clear, packet transmission is delayed until the FIFO contains the same number of bytes as the selected watermark. See "Ethernet General Control register (EGCR) bit definitions" on page 158 for more information.

Ethernet receiver considerations

When an Ethernet frame is received, DMA channel 1 searches the four buffer descriptors for the optimum buffer size. The search order is A, B, C, D. The search stops as soon as the DMA channel finds an available buffer that is large enough to hold the entire frame. The search also stops when the DMA channel finds a DMA Control register whose CE bit is set to zero.

Because interrupts are set when DMA channel 1 encounters buffers that are not ready, the device driver should be designed with the smallest buffers in the A pool and the largest buffers in the D pool. The number of available pools can be configured, from 1
to 4, with proper use of the CE bits.

An Ethernet receive FIFO overrun condition can occur (the FIFO becomes full while receiving an Ethernet packet) if insufficient buffers are allocated by the application. If this condition occurs (signaled by the NRIP bit in the DMA Channel 1 Status register), you must use this procedure to guarantee successful operation:

1. Set the ERXDMA, in the Ethernet General Control register, to zero.
2. SET the ERX bit, in the Ethernet General Control register, to zero.
3. Set the CE bit, in the DMA Control register, to zero.
4. Add new buffers for Ethernet DMA.
5. Restore the DMA Control register.
6. Restore the ERX bit.
7. Restore the ERXDMA bit.

External peripheral DMA support

DMA channels 3, 4, 5, and 6 can be set up for external DMA transfers, using three signals — DREQ_, DACK_, and DONE_ — to facilitate communications between the NS7520 and an external device. It is up to the external device to source or react to these signals. The external device can be a block of memory using memory-to-memory transfers. Within every transaction on the bus, a cycle on the external bus is executed corresponding to timing generated by the MEM module.

Signal description

Signal	Description
DREQ_	An input to the NS7520, sourced by the external device. All transfers are initiated when the external device asserts DREQ_ low. When the external device wants a DMA transfer (either read or write), it asserts the DREQ_ signal. DREQ_ can stay low until all data transfers are complete, or it can be removed as needed to control execution of the DMA.
DACK_	An input to the external device, sourced by the NS7520. When a bus transaction occurs, the NS7520 asserts DACK_; DACK_ may or may not return high between each data transfer. The external device can drive write data to the bus as soon as DACK_ is active.
DONE_IN	The external device asserts DONE_IN to retire the current DMA descriptor. When the NS7520 receives the DONE_IN signal, it closes the current buffer and sets the NCIP bit in the DMA Status register. The number of bytes that were received is in the buffer length field in the DMA buffer descriptor. DONE_IN can be asserted only when DACK_ is asserted.
DONE_OUT	The NS7520 asserts DONE_OUT when the L (Last) bit is set in the DMA buffer descriptor and the buffer is empty (retired).

External DMA configuration

The REQ bit in the DMA Control register must be set to 1 (external source) when configuring for external DMA, and the DMA output signals must be selected at GPIO PORTA. All other settings are the same as for any other DMA transfer.

The DMA channel 3/5 signals — DREQ1_, DACK1_, and DONE1_ — are multifunctional and shared with GPIO PORTA. The DMA channel 4/6 signals — DREQ2_, DACK2_, and DONE2_ — are multifunctional and shared with GPIO PORTC.

Memory-to-memory mode

In memory-to-memory mode, one of the source buffer pointers or destination buffer pointers points to the external device (accomplished by tying the device to a chip select). Each movement of data involves two autonomous operations: reading from the source and writing to the destination.

Figure 20 shows the signals needed for external memory-to-memory transfers.

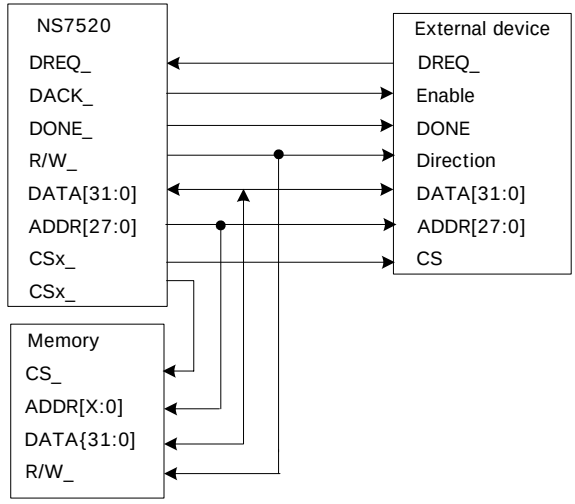


Figure 20: Hardware needed for external memory-to-memory DMA transfers

- If the source buffer pointer points to the external device, the device, in conjunction with appropriate memory device timing or external bus cycle timing, provides data when DACK_ is asserted.
- If the destination buffer pointer points to the external device, the device, in conjunction with appropriate memory device timing or external bus cycle timing, accepts data when DACK_ is asserted.
- TEA_ indicates the existence and termination of burst cycles.

DMA controller reset

You can simultaneously reset the DMA controllers for channels 1-13 without affecting any NS7520 modules. Just set the DMA reset bit in the GEN module System Control register to 1 and then back to 0 (see "DMARST" on page 67).

All DMA controllers are reset by all forms of hardware and software resets.

Ethernet Module

CHAPTER 9

The Ethernet controller module provides the NS7520 with one IEEE 802.3u compatible Ethernet interface. Two modules comprise the Ethernet interface: the Ethernet front-end (EFE) and the media access controller (MAC).

The MAC module interfaces to an external physical layer (PHY) device using the media independent interface (MII) standard defined by IEEE 802.3u. The MAC interface includes the MII clock and data signals.

Ethernet front-end (EFE)

Figure 21 shows a high-level block diagram of the EFE module. The EFE module provides the FIFO handling interface between the NS7520 BBus and MAC modules.

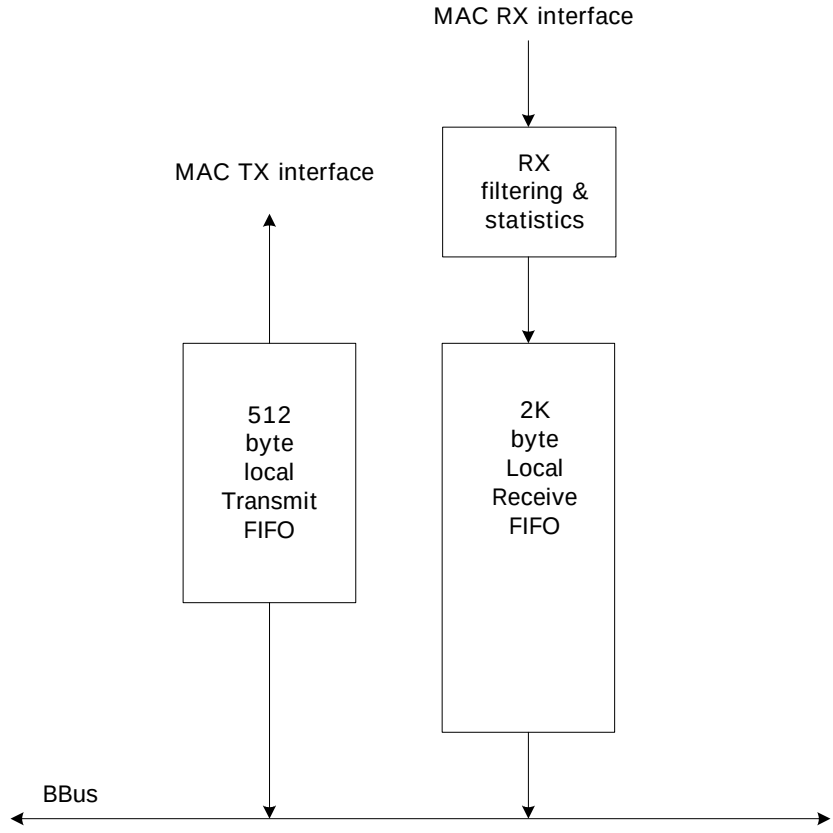


Figure 21: Ethernet front-end module

Basic features in the EFE include:

- Control and status registers for MAC, transmitter, and receiver
- 512-byte transmit FIFO
- 2048-byte local receive FIFO
- DMA interface logic

EFE logic provides all control and status registers required by the Ethernet module. The transmitter and receiver each provide a 16-bit status word after processing each Ethernet frame. These status words can be given to the CPU on an interrupt basis or moved automatically to the DMA buffer descriptor for the associated Ethernet frame.

Transmit and receive FIFOs

The EFE contains a 512-byte transmit FIFO and a 2048-byte local receive FIFO (storing filtered packets):

- **Transmit FIFO.** Allows the critical portion of the transmit buffer to wait in the FIFO while collisions occur on the Ethernet medium. This scheme removes the need for the transmitter to fetch the buffer multiple times from memory.
- **Receive FIFO.** Allows the entire Ethernet frame to be received and wait in the FIFO while the receive byte count is analyzed. The receive byte count is analyzed to determine the optimum buffer descriptor for DMA transfer. The DMA channel assigned to the Ethernet receiver can use one of four differently sized receive buffers. Only successfully received frames, with acceptable destination addresses, are committed to external system memory.

EFE transmit processing

The NS7520 Ethernet transmit DMA channel addresses one list of buffer descriptors per packet to be transmitted. The Ethernet transmit DMA channel moves Ethernet packets corresponding to this buffer descriptor list to the local FIFO in the EFE module.

EFE receive processing

The MAC block receives good Ethernet packets (those with a valid checksum and size); bad packets (those with invalid checksum, fragment errors, bad size, etc.) are discarded automatically.

The MAC forwards good Ethernet packets through address filtering. Packets can be filtered based on station address, broadcast, and select multicast packets. Those packets that pass through address filtering are moved into system memory.

Receive buffer descriptor selection

When an Ethernet frame is received, the Ethernet receiver DMA channel searches the four buffer descriptors — starting with A and continuing with B, C, and D, as necessary — for an appropriately sized buffer. The search stops when the DMA channel finds an available buffer that is large enough to hold the entire Ethernet receive frame. The search also stops when the DMA channel finds a Control register whose Channel Enable (CE) bit is zero.

Figure 22 shows receiver buffer size selection.

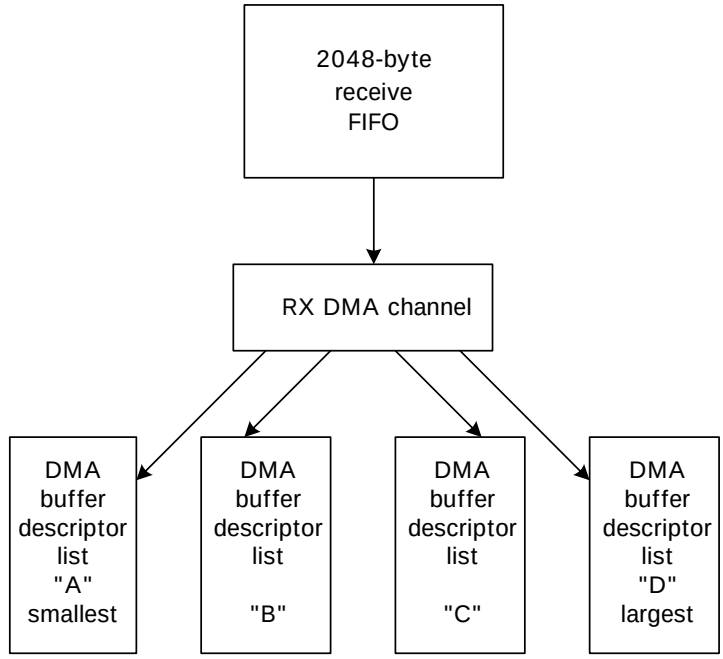


Figure 22: Receive buffer size selection

Interrupts are set when the DMA channel encounters buffers that are *not ready*. The device driver should be designed with the smallest buffers in the A pool and the largest buffers in the D pool. The number of available pools can be configured (from 1 to 4) using the channel enable bits in the DMA Control register (see "DMA Control register" on page 136).

External CAM filtering

The NS7520 provides support for external CAM filtering, which requires an external CAM controller to operate in conjunction with the MAC within the NS7520.

To support CAM filtering, the PORTC2 and PORTC5 signals must be configured for special function signals RPSF_ and REJECT_, respectively. See "PORTC Configuration register," beginning on page 77, for information.

The NS7520 drives the RPSF_ signal active low to identify when each Ethernet packet starts to be transferred from the Ethernet PHY to the NS7520 internal MAC; the signal is driven low while the fifth nibble is being transferred. The external CAM hardware monitors the MII receive interface between the PHY and the MAC, waiting for the RPSF_ assertion. When RPSF_ is asserted, the CAM hardware can extract the destination address field from the MII receive bus.

After performing destination address lookup, the CAM filtering hardware can reject the incoming packet by asserting the REJECT_ input active low during any nibble time between RPSF_ assertion and nibble number 128.

"Ethernet cam timing" on page 297 shows the timing relationship between the RPSF_, REJECT_, and MII receive interface signals. The MII receive interface is transferring a packet whose first six nibbles have the values 1, 2, 3, 4, 5, and 6. The RPSF_ signal is activated while the fifth nibble is being transferred from MII. Nibbles are transferred in Little Endian order within a byte.

The external CAM hardware uses the RPSF_ signal to find the alignment for the destination address. After the lookup is performed, the CAM hardware can assert the REJECT_ signal to discard the frame. The REJECT_ signal can be asserted during any RXCLK cycle between when RPSF_ is asserted and nibble number 128.

MAC module

The MAC component of the NS7520 provides a full function 10/100 Mbps media access controller module with media independent interface (MII) and optional interface modules, which include MII, PMD, and ENDEC.

The basic features in the MAC module include:

- 10/100 Mbps Ethernet MAC
- MII interface
- MII management function
- Optional 100 Mbps physical coding sublayer (PCS)
- Optional 10 Mbps ENDEC interface
- Address filtering
- Statistics gathering

The 10/100 Mbps Ethernet MAC can be connected to the following devices through the MII:

- 10/100 Mbps PHY
- 100 Mbps TP-PMD
- 10 Mbps ENDEC

The MAC module also provides MII management and station address logic (SAL). The SAL block filters the incoming receive addresses, accepting or rejecting individual and multicast receive packets.

MAC module block diagram

Figure 23 shows the MAC block and associated hierarchy, containing three main modules:

- **TFUN.** Transmit function.
- **RFUN.** Receive function.
- **MCS.** MAC control sublayer.

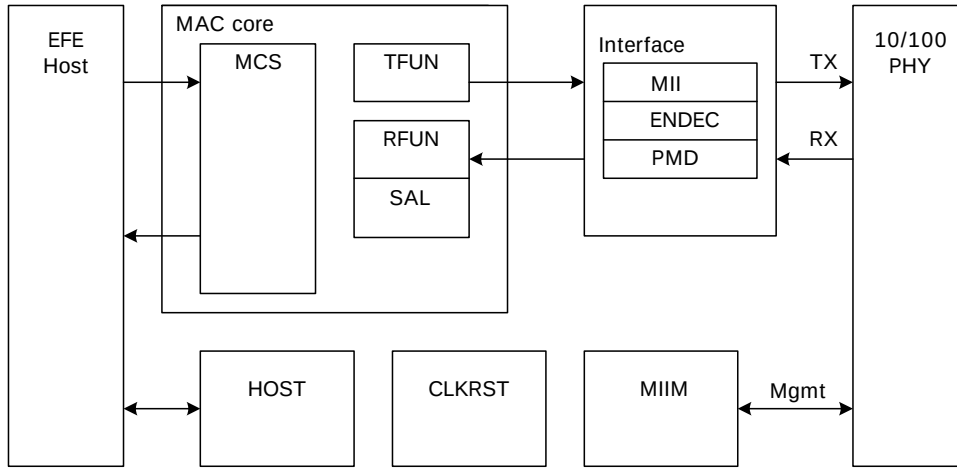


Figure 23: MAC block diagram

Other modules in the diagram include:

- **MAC core — 10/100 Mbps media access controller.** Performs the CSMA/CD function and flow control functions. See ISO/IEC and IEEE Standard 802.3x documentation for more information.
- **Host — Host interface.** Provides a 16-bit interface for control and configuration.
- **CLKRST — Clocks and resets.** Provides a central location for clock trees and reset logic.
- **MIIM — MII management.** Provides a control/status path to MII PHYs.
- **SAL — Station address logic.** Performs station address, broadcast, and multicast filtering.

DMA channel assignments

One DMA channel is dedicated to Ethernet receive and one DMA channel is dedicated to Ethernet transmit. The Ethernet receiver has four DMA subchannels, which support the receive buffer descriptor selection feature (see "DMA buffer descriptor," beginning on page 130).

- The Ethernet receiver is assigned DMA channel 1 (1A, 1B, 1C, and 1D).
- The Ethernet transmitter is assigned DMA channel 2.

EFE configuration

Table 52 shows the Ethernet front-end register map. All registers are 32 bits unless otherwise noted.

Note: Reading or writing the MAC configuration registers (address locations 0xFF80 0400 through 0xFF80 05DC) is unreliable without valid clocks on the TXCLK and RXCLK input pins.

Address	Register	Register description
FF80 0000	EGCR	Ethernet General Control register
FF80 0004	EGSR	Ethernet General Status register
FF80 0008	FIFO	Ethernet FIFO Data register
FF80 000C	FIFOL	Ethernet FIFO Data Register Last
FF80 0010	ETSR	Ethernet Transmit Status register
FF80 0014	ERSR	Ethernet Receive Status register
FF80 0400	MAC1	MAC Configuration Register 1
FF80 0404	MAC2	MAC Configuration Register 2
FF80 0408	IPGT	Back-to-Back Inter-Packet-Gap register
FF80 040C	IPGR	Non-Back-to-Back Inter-Packet-Gap register
FF80 0410	CLRT	Collision Window/Retry register

Table 52: EFE register map

Address	Register	Register description
FF80 0414	MAXF	Maximum Frame register
FF80 0418	SUPP	PHY Support register
FF80 041C	TEST	Test register
FF80 0420	MCFG	MII Management Configuration register
FF80 0424	MCMD	MII Management Command register
FF80 0428	MADR	MII Management Address register
FF80 042C	MWTD	MII Management Write Data register
FF80 0430	MRDD	MII Management Read Data register
FF80 0434	MIND	MII Management Indicators register
FF80 0438	SMII	SMII Status register
FF80 0440	SA1	Station Address Register 1
FF80 0444	SA2	Station Address Register 2
FF80 0448	SA3	Station Address Register 3
FF80 05C0	SAFR	Station Address Filter register
FF80 05D0	HT1	Hash Table Register 1
FF80 05D4	HT2	Hash Table Register 2
FF80 05D8	HT3	Hash Table Register 3
FF80 05DC	HT4	Hash Table Register 4

Table 52: EFE register map

Ethernet General Control register (EGCR) bit definitions

Address: FF80 0000

General information

These fields should be set only once, on device open:

- ERX
- ERXDMA
- ERXLNG
- ERXSHT
- ERXBAD
- ETX
- ETXDMA
- ETXWM
- EFULLD

These fields are used *only* when using Ethernet receive in interrupt service mode rather than DMA mode (DMA interface logic). DMA mode provides higher performance out of the Ethernet chip, and can be turned on and off.

- ERXREG
- ERFIFOH
- ERXBR
- ETXREG
- ETFIFOH
- ETXBC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERX	ERX DMA	ERX LNG	ERX SHT	ERX REG	ER FIFOH	ERX BR	ERX BAD	ETX	ETX DMA	ETXWM		ETX REG	ET FIFOH	ETX BC	E FULLD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE		Rsvd	RXC INV	TXC INV	pNA	MAC_ RESET	ITXA	PDN:, AUI_TP:, LNK_DIS:, LPBK:, UTP_STP						EXINT	

Register bit assignment

Note: Bits D15:D00 are media control bits, with D07:D00 used in ENDEC mode only.

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ERX	0	**Enable receive FIFO** 0 Disables inbound data flow and resets the FIFO 1 Enables inbound data flow Set to 1 to allow data to be received from the MAC receiver. Set to 0 (clear) to reset the receive side FIFO.
D30	R/W	ERXDMA	0	**Enable receive DMA** 0 Disables inbound DMA data request 1 Enables inbound DMA data request Set to 1 to allow the EFE module to issue receive data move requests to the DMA controller. Clear this bit to temporarily stall receive side Ethernet DMA.
D29	R/W	ERXLNG	0	**Accept long (> 1520 bytes [MAXF setting]) receive packets** When set to 1, allows the MAC to accept packets that are larger than 1520 bytes.
D28	R/W	ERXSHT	0	**Accept short (< 60 bytes) receive packets** When set to 1, allows the MAC to accept packets that are smaller than 60 bytes. The ERXSHT bit is used primarily for debugging.
D27	R/W	ERXREG	0	**Enable Receive Data register ready interrupt** Set to 1 to generate an interrupt when data is available in the RX FIFO.
D26	R/W	ERFIFOH	0	**Enable receive data FIFO half full interrupt** Set to 1 to generate an interrupt when the RX FIFO is at least half full (1024 bytes).
D25	R/W	ERXBR	0	**Enable receive buffer ready interrupt** Set to 1 to generate an interrupt when a new data packet is available in the RX FIFO.

Table 53: Ethernet General Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D24	R/W	ERXBAD	0	**Accept bad receive packets** When set to 1, allows the MAC to accept packets received in error. Bad receive packets include those packets with CRC errors, alignment errors, and dribble errors. The ERXBAD bit is used primarily for debugging.
D23	R/W	ETX	0	**Enable transmit FIFO** 0 Disables outbound data flow and resets the FIFO 1 Enables outbound data flow Set to 1 to allow data to be written to the TX FIFO. Clear to reset the transmit side FIFO.
D22	R/W	ETXDMA	0	**Enable transmit DMA** 0 Disables outbound DMA data request 1 Enables outbound DMA data request Set to 1 to allow the EFE module to issue transmit data move requests to the DMA controller. Clear this bit to temporarily stall transmit side Ethernet DMA. **Do *not* set this bit when operating the Ethernet receiver in interrupt service mode.**
D21:20	R/W	ETXWM	0	**Transmit FIFO water mark before transmit start** 00 25% FIFO full 01 50% FIFO full 10 75% FIFO full 11 Reserved Identifies the minimum number of bytes required in the transmit FIFO to initiate packet transmission. A larger watermark setting increases transmit packet latency, allowing for more slack in the memory system FIFO fill rate. Note that packet transmission can also be initiated using DMA (see "Ethernet transmitter considerations" on page 144) and the FIFO Data register (see "Ethernet FIFO Data register" on page 167).

Table 53: Ethernet General Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D19	R/W	ETXREG	0	**Enable Transmit Data register ready interrupt** Set to 1 to generate an interrupt when the TX FIFO is ready to accept data.
D18	R/W	ETFIFOH	0	**Enable transmit data FIFO half empty interrupt** Must be set to 1 to generate an interrupt when the TX FIFO is at least half empty (< 256 bytes).
D17	R/W	ETXBC	0	**Enable transmit buffer complete interrupt** Set to 1 to generate an interrupt when the transmit packet transmission is complete.
D16	R/W	EFULLD	0	**Enable full-duplex operation** Set to 1 to allow the Ethernet TX and RX DMA operations to operate simultaneously. This bit must be set when the Ethernet link negotiation results in full-duplex mode. It is recommended that you limit transmit frames to 512 bytes when full-duplex operation is enabled.
D15:14	R/W	MODE	0	**Ethernet interface mode** 00 10/100 Mbps MII mode 10 10 Mbps Level 1 ENDEC mode 11 10 Mbps SEEQ ENDEC mode Identifies the type of Ethernet PHY attached to the NS7520. *In ENDEC SEEQ mode*, the LPBK pin is inverted before driving the MDC pin. *In ENDEC Level 1 mode*, the PDN signal is driven using an open-drain output driver.
D13	N/A	Reserved	N/A	Reads as zero.
D12	R/W	RXCINV	0	**Invert the receive clock input** Set to 1 only when the external Ethernet PHY generates a clock that is inverted in phase relative to what the MAC is expecting. This bit is not required for most commercially available PHY devices.

Table 53: Ethernet General Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D11	R/W	TXCINV	0	**Invert the transmit clock input** Set to 1 only when the external Ethernet PHY generates a clock that is inverted in phase relative to what the MAC is expecting. This bit is not required for most commercially available PHY devices.
D10	R/W	pNA	0	**pSOS pNA buffer descriptors** 0 Standard receiver format. The data block immediately follows the 14-byte header block. 1 pSOS pNA receiver format. The receiver inserts a 2-byte padding between the 14-byte header and the data block. This configuration aligns both the header and the data blocks on a 32-bit longword boundary.
D09	R/W	MAC_RESET	0	**MAC software reset** 0 Restore MAC to normal operation 1 Reset MAC host interface
D08	R/W	ITXA	0	**Insert transmit source address** When set, forces the MAC to automatically insert the Ethernet source MAC address into the Ethernet transmit packet. The MAC address information is provided by the data configured in the Station Address registers (SA1, SA2, SA3). When ITXA is cleared, the 7th through 12th bytes in the Ethernet packet are ignored and replaced by the size bytes in the MAC Address register.
D07:02	R/W	PDN: AUI_TP: LNK_DIS: LPBK: UTP_STP	0	**ENDEC media control bits** Used only when the MODE field is configured for ENDEC mode. In this configuration, each register bit is mapped to NS7520 pins, allowing the software to manipulate control signals on an external ENDEC PHY device. See Table 54 on page 163 for more information.

Table 53: Ethernet General Control register bit definition

Bits	Access	Mnemonic	Reset	Description
D01:00	R/W	EXINT	0	**External interface mode** 00 MII normal operation, used for all MII-style 10/100 PHY devices 01 TP-PMD mode, used for PHYs that contain their own non-standard PCS circuitry 10 10 Mbit mode, used for older 10-Mbit-only PHY devices that predate the MII standard. 11 Reserved

Table 53: Ethernet General Control register bit definition

ENDEC mode and NS7520 pins

Table 54 shows the relationship between the lower bits in the Ethernet General Control register and the NS7520 pins that they control. The NS7520 pins are controlled by these bits only when the MODE field (D15:14) is set to ENDEC.

MODE field	Output based on EFE CSR bit
Not ‘b00	TXD1 = PDN inverted, open drain
Not ‘b00	TXD2 = AUI_TP[1]
Not ‘b00	TXD3 = AUI_TP[0]
Not ‘b00	TXER = LNK_DIS_
Not ‘b00 and not ‘b11	MDC = LPBK
‘b11	MDC = LPBK inverted
Not ‘b00	MDO = UTP_STP

Table 54: ENDEC control signal cross-reference

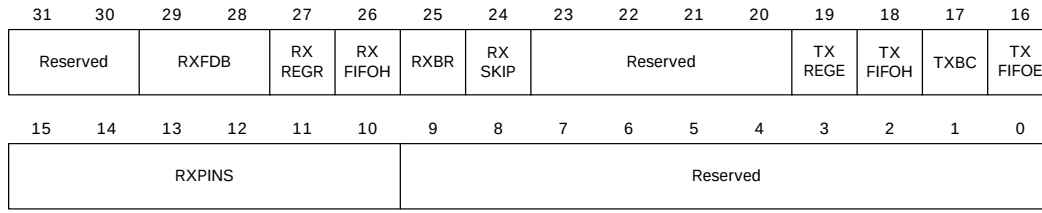
Ethernet General Status register (EGSR) bit definitions

Address: FF80 0004

General information

These fields are used *only* when using Ethernet receive in interrupt service mode rather than DMA mode (DMA interface logic). DMA mode provides higher performance out of the Ethernet chip, and can be turned on and off.

- RXFDB
- RXREGR
- RXFIFOH
- RXBR
- RXSKIP
- TXREGE
- TXFIFOH
- TXBC



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:30	N/A	Reserved	N/A	N/A

Table 55: Ethernet General Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D29:28	R	RXFDB	0	**Receive FIFO data available** Valid only when RXREGR (D27) = 1. 00 Full-word 01 One byte 10 Half-word 11 Three bytes; LENDIAN determines which three Must be used in conjunction with RXREGR. When RXREGR is set to 1, RXFDB identifies how many bytes are available in the Receive Data register.
D27	R	RXREGR	0	**Receive register ready** Set to 1 when data is available to be read from the FIFO Data register. When set active high, this bit can cause an interrupt when the ERXREG bit is also set (in the Ethernet General Control register). RXREGR is never active when RXBR (D25) is set; RXBR must be cleared to activate RXREGR.
D26	R	RXFIFOH	0	**Receive FIFO half full** Set to 1 when the receive FIFO is at least half full (> 1024 bytes). When set active high, this bit can cause an interrupt when the ERFIFOH bit is set (in the Ethernet General Control register).
D25	R/C	RXBR	0	**Receive buffer ready** Set to 1 when a new packet is available in the receive FIFO. When set active high, this bit can cause an interrupt when the ERXBR bit is also set (in the Ethernet General Control register). RXBR indicates to the interrupt service routine (ISR) that the Receive Status register should be read. After reading that register, clear RXBR by writing a 1 to the RXBR position in this (Ethernet General Status) register. When RXBR is cleared, RXREGR and RXFIFOH become active.

Table 55: Ethernet General Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D24	R/W	RXSKIP	0	**Receive buffer skip** Used (written) instead of clearing the RXBR bit. Writing to RXSKIP clears the RXBR bit and flushes the next available packet from the receive FIFO. Using RXSKIP is a means of performing receive packet filtering in the interrupt service routine; the packet simply is flushed from the FIFO rather than read out of the FIFO.
D23:20	N/A	Reserved	N/A	N/A
D19	R	TXREGE	0	**Transmit register empty** Set to 1 whenever the transmit FIFO is ready to accept data. When active high, this bit can cause an interrupt when the ETXREGE bit is also set (in the Ethernet General Control register). TXREGE is never active when TXBC (D17) is set; TXBC must be cleared to activate TXREGE.
D18	R	TXFIFOH	0	**Transmit FIFO half empty** Set to 1 when the transmit FIFO is at least half empty. When active high, this bit can cause an interrupt when the ETFIFOH bit is also set (in the Ethernet General Control register).
D17	R/C	TXBC	0	**Transmit buffer complete** Set when packet transmission is complete. When active high, this bit can cause an interrupt when the EXTBC bit is also set (in the Ethernet General Control register). The TXBC bit indicates to the interrupt service routine to read the Ethernet Transmit Status register. After the Ethernet Transmit Status register is read, clear the TXBC bit by writing a 1 to the TXBC bit position in this (Ethernet General Status) register. After TXBC is cleared, the TXREGE and TXFIFOH bits become active.
D16	R	TXFIFOE	1	**Transmit FIFO empty** Active (set to 1) when the transmit FIFO is empty.

Table 55: Ethernet General Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D15:10	R	RXPINS	0	**ENDEC PHY status** The logic state of the pins RXD2, RXD1, RXD3, RXER, and RXDV is read from this field. These pins are useful only with an ENDEC PHY, and monitor status signals on the PHY. When an MII PHY is used, these bits should be ignored. See Table 56 on page 167 for more information.
D09:00	N/A	Reserved	N/A	N/A

Table 55: Ethernet General Status register bit definition

Table 56 shows the relationship between the lower bits in the Ethernet General Status register and the NS7520 pins they monitor. The significance of each bit depends on the selected PHY and the wiring of the PHY to these pins.

Data bit	NS7520 pin
D15	RXD2
D13	RXD1
D12	RXD3
D11	RXER
D10	RXDV

Table 56: ENDEC status signal cross-reference

Ethernet FIFO Data register

Address: FF80 0008 / FF80 000C (secondary address)

The Ethernet FIFO Data register allows manual interface with the Ethernet FIFO, rather than using DMA support. This register is used primarily as a diagnostic tool.

Writing to the Ethernet FIFO Data register

Writing to the Ethernet FIFO Data register loads the transmit FIFO. This register can be written only when the TXREGE bit is set in the Ethernet General Status register, indicating that space is available in the transmit FIFO.

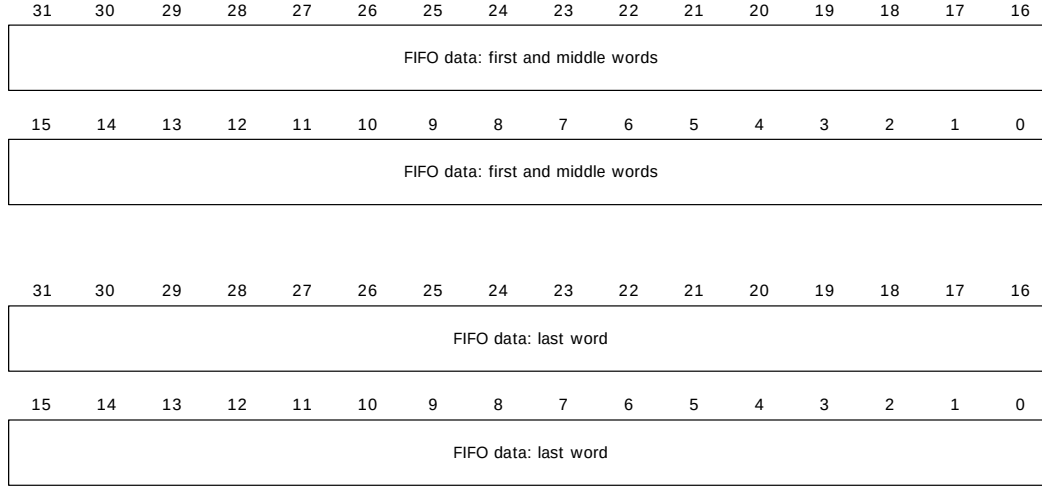
The transmit FIFO has a secondary address (FF80 000C) that signifies the last word of a transmit frame. The first and middle words must use the primary address (FF80 0008).

Writing to the secondary address with the transmit interrupts disabled (ETXBC in the Ethernet General Control register) initiates transmission of data from the transmit FIFO. Otherwise, transmission begins when the TX FIFO byte count equals the selected watermark (ETXWM in the Ethernet General Control register).

Reading from the Ethernet FIFO Data register

Reading from the Ethernet FIFO Data register empties the receive FIFO. The Ethernet General Status register indicates how many bytes are available to be read. The receive FIFO is available only when the RXBR bit has been set to 1, which clears the bit, in the Ethernet General Status register.

Note: The Ethernet Receive Status register (see "Ethernet Receive Status register" on page 174) should be read before clearing the RXBR bit. When operating in interrupt service mode, RXBR is cleared manually. When operating in DMA mode, the Ethernet Receive Status register is read automatically and RXBR is cleared automatically.



Register bit assignment

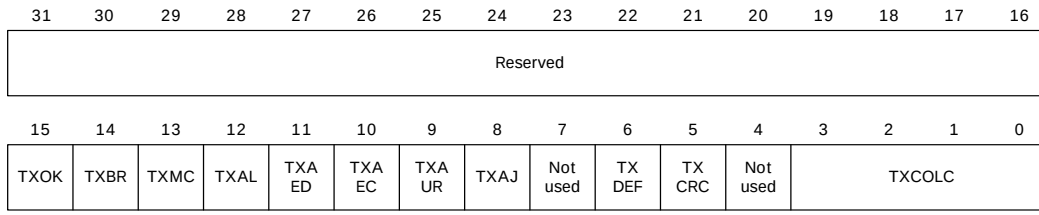
Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	FIFO	N/A	**FIFO data — FF80 0008** First and middle words
D31:00	R/W	FIFO	N/A	**FIFO data — FF80 000C** Last word, write-only

Table 57: Ethernet FIFO Data register bit definition

Ethernet Transmit Status register

Address: FF80 0010

The Ethernet Transmit Status register contains the status for the last completed transmit buffer. The transmit buffer complete bit (TXBC) is set in the Ethernet General Status register when a transmit frame is completed and the Ethernet Transmit Status register is loaded. The lower 16 bits (D15:00) of the register are also loaded into the StatusOrIndex field of the DMA buffer descriptor when using DMA mode.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A

Table 58: Ethernet Transmit Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D15	R	TXOK	**0**	**Packet transmitted OK** ■ Set to 1 when the last Ethernet packet was transmitted without error. ■ Set to 0 when the last Ethernet packet was not transmitted. When the bit is set to 1, the frame has been delivered and emptied from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXOK is set.
D14	R	TXBR	**0**	**Broadcast packet transmitted** Set to 1 to indicate that the last Ethernet packet transmitted was a broadcast packet.
D13	R	TXMC	**0**	**Multicast packet transmitted** Set to 1 to indicate that the last Ethernet packet transmitted was a multicast packet.
D12	R	TXAL	**0**	**Transmit abort — late collision** Set to 1 to indicate that the last Ethernet packet was not transmitted successfully; packet transmission was aborted due to a late collision problem. When this bit is set, the transmit frame is automatically flushed from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXAL is set.

Table 58: Ethernet Transmit Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D11	R	TXAED	**0**	**Transmit abort — excessive deferral** Set to 1 to indicate that the last Ethernet packet was not transmitted successfully; packet transmission was aborted due to *excessive deferral*. Excessive deferral means that there was a receive carrier on the Ethernet channel for a long period of time, making it impossible for the transmitter to try to transmit. When this bit is set, the transmit frame is automatically flushed from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXAED is set.
D10	R	TXAEC	**0**	**Transmit abort — excessive collisions** Set to 1 to indicate that the last Ethernet packet was not transmitted successfully; packet transmission was aborted due to an excessive number of collisions. The maximum number of collisions allowed is determined by the RETRY field in the Collisions Window/Collision Retry register. When this bit is set, the transmit frame is automatically flushed from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXAEC is set.

Table 58: Ethernet Transmit Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D09	R	TXAUR	**0**	**Transmit aborted — underrun** Set to 1 to indicate that the last Ethernet packet was not transmitted successfully; packet transmission was aborted due to a FIFO *underrun* condition. A FIFO underrun condition indicates that the DMA controller was unable to fill the FIFO at a fast enough rate compared to the rate of transmission on the Ethernet medium, for one of these reasons: ■ The DMA controller was not configured for bursting. ■ The memory peripheral device was not configured for bursting. ■ The memory peripheral device is too slow to support the Ethernet interface. When this bit is set, the transmit frame is flushed automatically from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXAUR is set.
D08	R	TXAJ	**0**	**Transmit abort — jumbo** Set to 1 to indicate that the last Ethernet packet was not transmitted successfully; packet transmission was aborted due to a *jumbo* condition. A jumbo condition means that the packet was too large; that is, greater than 1518 bytes. Packets larger than 1518 bytes are not transmitted successfully unless the HUGEN bit is set in the MAC Configuration register. When this bit is set, the transmit frame is automatically flushed from the transmit FIFO. TXREGE and TXFIFOH in the Ethernet General Status register become active when the FIFO is ready to start receiving the next packet. TXBC in the Ethernet General Status register becomes active when TXAJ is set.
D07	N/A	Not used	N/A	Always set to 0.

Table 58: Ethernet Transmit Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D06	R	TXDEF	**0**	**Transmit packet deferred** Set to 1 to indicate that the last successfully transmitted Ethernet packet encountered a deferral. Transmission was delayed because the Ethernet medium was busy when trying to send the first transmission. When this bit is set (instead of TXOK or any of the TX abort bits), the transmitter tries to resend the packet. This process continues until the packet is transmitted successfully or an abort condition is detected. A change in transmit error bits can be determined only by polling.
D05	R	TXCRC	**0**	**Transmit CRC error** Set to 1 to indicate that the last successfully transmitted Ethernet packet had an embedded CRC error. This condition occurs only when the CRCEN bit in the MAC Configuration register is set to 0. When CRCEN is set to 0, the MAC does not insert a CRC; the MAC expects a precompiled CRC to be contained in the last four bytes of the Ethernet packet. If the MAC finds that the precompiled CRC is incorrect, the MAC sets the TXCRC bit in this (Ethernet Transmit Status) register. When this bit is set (instead of TXOK or any of the TX abort bits), the transmitter tries to resend the packet. This process continues until the packet is transmitted successfully or an abort condition is detected. A change in transmit error bits can be determined only by polling.
D04	N/A	Not used	0	Always 0.

Table 58: Ethernet Transmit Status register bit definition

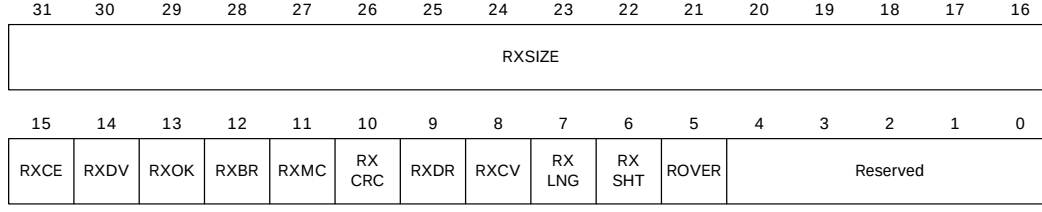
Bits	Access	Mnemonic	Reset	Description
D03:00	R	TXCOLC	**0**	**Transmit collision count** Indicates how many collisions the MAC encountered while it was trying to transmit the package. TXCOLC indicates only that collision events have occurred; if the packet transmission was aborted, the TXAEC bit will be set. The MAC tries to retransmit the packet up to the maximum number of collision retries defined by the RETRY field in the Collision Window/Collision Retry register. This bit is valid (1) when the TXAL or TXAEC bits are set; otherwise, the bit is set to 0.

Table 58: Ethernet Transmit Status register bit definition

Ethernet Receive Status register

Address: FF80 0014

The Ethernet Receive Status register contains the status for the last completed receive buffer. The receive buffer complete bit (RXBR) is set in the Ethernet General Status register when a receive frame is completed and the Receive Status register is loaded. The lower 16 bits (D15:00) of the register are also loaded into the StatusOrIndex field of the DMA buffer descriptor when using DMA mode.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	R	RXSIZE	**0**	**Receive buffer size in bytes** Provides the number of bytes contained in the next packet to be read from the Ethernet receive FIFO. When using DMA mode, the RXSIZE field is also copied to the buffer length field in the DMA buffer descriptor.
D15	R	RXCE	**0**	**Receive carrier event previously seen** Set to 1 to indicate that the MAC received a *carrier event* from the Ethernet PHY since the last time a receive packet event was recorded in this status register. The carrier event is not associated with this particular packet. A carrier event is defined as any activity on the receive channel that does not result in a packet receive attempt.
D14	R	RXDV	**0**	**Receive data violation event previously seen** Set to 1 to indicate that, at some point since the last recorded receive packet, a *receive data violation* was detected, noted, and reported with this receive packet event. The receive data event is not associated with this particular packet. A receive data violation event occurs when there is no valid preamble and start frame delimiter (SFD) in the data stream.
D13	R	RXOK	**0**	**Receive packet OK** Set to 1 to indicate that the next packet in the receive FIFO has been received without any errors. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register become active and the packet can be emptied (using interrupts, polling, or DMA). When the packet has been emptied, RXREGR and RXFIFOH become inactive until the next receive buffer is ready.

Table 59: Ethernet Receive Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D12	R	RXBR	**0**	**Receive broadcast packet** Set to 1 to indicate that the next packet in the receive FIFO is a broadcast packet. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register become active and the packet can be emptied (using interrupts, polling, or DMA). When the packet has been emptied, RXREGR and RXFIFOH become inactive until the next receive buffer is ready.
D11	R	RXMC	**0**	**Receive multicast packet** Set to 1 to indicate that the next packet in the receive FIFO is a multicast packet. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register become active and the packet can be emptied (using interrupts, polling, or DMA). When the packet has been emptied, RXREGR and RXFIFOH become inactive until the next receive buffer is ready.
D10	R	RXCRC	**0**	**Receive packet has CRC error** Set to 1 to indicate that the next packet in the receive FIFO was received with a CRC error. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.
D09	R	RXDR	**0**	**Receive packet has dribble bit error** Set to 1 to indicate that, at the end of the next packet in the receive FIFO, an additional 1 to 7 bits of data (dribble nibble) were received. **Note:** This packet is considered valid if the RXCRC bit (D10) is not set. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.

Table 59: Ethernet Receive Status register bit definition

Bits	Access	Mnemonic	Reset	Description
D08	R	RXCV	**0**	**Receive packet has code violation** Set to 1 to indicate that the Ethernet PHY asserted the RXER signal during the data phase of this frame. RXCV indicates that the PHY encountered invalid receive codes while receiving the data. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.
D07	R	RXLNG	**0**	**Receive packet is too long** Set to 1 to indicate that the next packet in the receive FIFO is longer than 1518 bits. A long packet is accepted only when the ERXLNG bit is set in the Ethernet General Control register. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.
D06	R	RXSHT	**0**	**Receive packet is too short** Set to 1 to indicate that the next packet in the receive FIFO is smaller than 64 bytes. A short packet is accepted only when the ERXSHT bit is set in the Ethernet General Control register. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.

Table 59: Ethernet Receive Status register bit definition

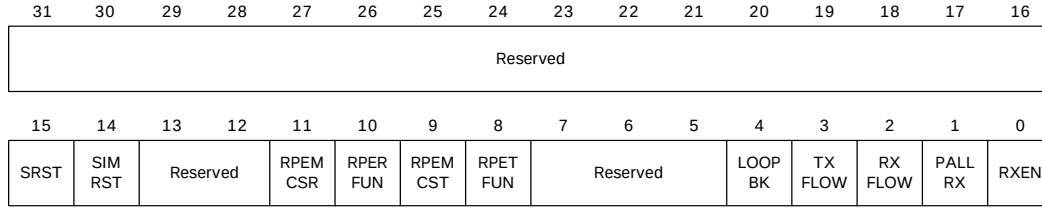
Bits	Access	Mnemonic	Reset	Description
D05	R	ROVER	**0**	**Receive overflow** Set to 1 to indicate that a receive FIFO *overrun* condition has occurred. An overrun condition occurs when the FIFO becomes full while receiving an Ethernet packet. This condition indicates that the DMA controller was not able to empty the FIFO at a fast enough rate compared to the rate that information was received from the Ethernet medium for one of these reasons: ■ The DMA controller was not configured for bursting. ■ The memory peripheral device was not configured for bursting. ■ The memory peripheral device is too slow to support the Ethernet interface. When this bit is set, the RXREGR and RXFIFOH bits in the Ethernet General Status register remain inactive. The bad receive packet is flushed immediately from the FIFO.
D04:00	N/A	Reserved	N/A	N/A

Table 59: Ethernet Receive Status register bit definition

MAC Configuration Register 1

Address: FF80 0400

MAC Configuration Register 1 contains bits that control functionality within the Ethernet MAC block.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	SRST	1	**Soft reset** Must be set before/while changing any other MAC bits, and cleared after they are modified. Setting this bit puts all MAC modules within the MAC block into reset mode, with the exception of the host interface. The host interface can be reset only by a hardware reset condition.
D14	R/W	SIMRST	0	**Simulation reset** Set to 1 to reset the random number generator within the MAC transmitter.
D13:12	N/A	Reserved	N/A	N/A
D11	R/W	RPEMCSR	0	**Reset PEMCS/RX** Set to 1 to reset the MAC control sublayer/receive domain logic.
D10	R/W	RPERFUN	0	**Reset PERFUN** Set to 1 to reset the MAC receive function logic.
D09	R/W	RPEMCST	0	**Reset PEMCS/TX** Set to 1 to reset the MAC control sublayer/transmit domain logic.
D08	R/W	RPETFUN	0	**Reset PETFUN** Set to 1 to reset the MAC transmit function logic.
D07:05	N/A	Reserved	N/A	N/A
D04	R/W	LOOPBK	0	**Internal loopback** 1 The MAC transmit interface loops back to the MAC receive interface. 0 (clear) Normal operation.
D03	R/W	TXFLOW	0	**TX flow control** 1 Allows the MAC to transmit PAUSE flow control frames 0 Blocks PAUSE flow control frame transmission

Table 60: MAC Configuration Register 1 bit definition

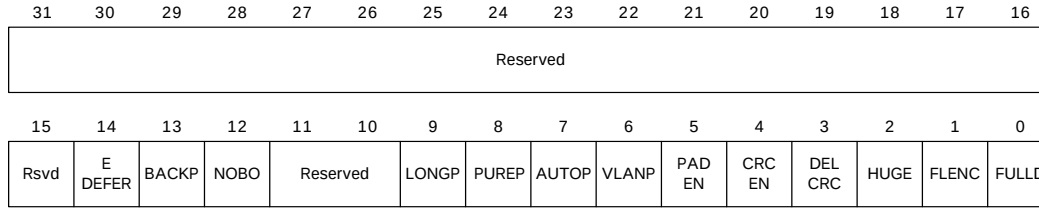
Bits	Access	Mnemonic	Reset	Description
D02	R/W	RXFLOW	0	**RX flow control** 1 The MAC acts on received PAUSE flow control frames. 0 The MAC ignores all PAUSE flow control frames.
D01	R/W	PALLRX	0	**Pass ALL receive frames** 1 The MAC receiver indicates PASS CURRENT RECEIVE FRAME for all frames, no matter the type (normal or flow control). 0 Deasserts PASS CURRENT RECEIVE FRAME for valid control frames.
D00	R/W	RXEN	0	**Receive enable** Set to 1 to allow the MAC receiver to receive frames. Internally, the MAC synchronizes this control bit to the incoming receive stream and outputs SYNCHRONIZED RECEIVE ENABLE. The host system uses SYNCHRONIZED RECEIVE ENABLE to qualify receive frames.

Table 60: MAC Configuration Register 1 bit definition

MAC Configuration Register 2

Address: FF80 0404

MAC Configuration Register 2 contains bits that control functionality within the Ethernet MAC block.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:15	N/A	Reserved	N/A	N/A
D14	R/W	EDEFER	**0**	**Excess Deferral** 1 Allows the MAC to defer to carrier indefinitely, per the 802.3u standard. 0 The MAC aborts when the excessive deferral limit is reached. The MAC provides feedback to the host system (through a transmit abort condition defined in the Transmit Status register (see "TXAED" on page 171).
D13	R/W	BACKP	0	**Backpressure/NO back off** 0 After a collision, the MAC waits a random amount of time to retransmit packets (“back off”). 1 After a collision, the MAC immediately retransmits packets without backoff. This reduces the chance of further collisions and ensures that transmit packets are sent.
D12	R/W	NOBO	0	**No back off** When this bit is set to 1, the MAC immediately retransmits after a collision, rather than using the binary exponential back off algorithm defined in the 802.3u standard.
D11:10	N/A	Reserved	N/A	N/A
D09	R/W	LONGP	0	**Long preamble enforcement** 1 The MAC allows only those receive packets that contain preamble fields less than 12 bytes in length. 0 The MAC allows any length preamble, as defined in the 802.3u standard.
D08	R/W	PUREP	0	**Pure preamble enforcement** 1 The MAC verifies the preamble‘s content to ensure it contains `'h55` and is error-free. A packet with errors in the preamble is discarded. 0 There is no preamble checking.

Table 61: MAC Configuration Register 2 bit definition

Bits	Access	Mnemonic	Reset	Description
D07	R/W	AUTOP	0	**Auto detect pad enable** When set to 1, the MAC automatically detects the frame type — tagged or untagged — by comparing the two octets following the source address with `'h8100` VLAN protocol ID and pad accordingly. See Table 62: "PAD operation" on page 183 for additional information. **Note:** This bit is ignored if PADEN (D05) is set to 0.
D06	R/W	VLANP	0	**VLAN pad enable** When set to 1, the MAC pads all short frames to 64 bytes and appends a valid CRC. See Table 62: "PAD operation" on page 183 for additional information. **Note:** This bit is ignored if PADEN (D05) is set to 0.
D05	R/W	PADEN	0	**PAD/CRC enable** 1 The MAC pads all short frames. 0 Frames presented to the MAC have a valid length. PADEN is used in conjunction with the Auto detect pad enable and VLAN pad enable fields. See Table 62: "PAD operation" on page 183 for additional information.
D04	R/W	CRCEN	0	**CRC enable** 1 Appends a CRC to every frame whether or not padding is required. This bit must be set to 1 if PAD/CRC is enabled. 0 Indicates that frames presented to the MAC already contain a CRC.
D03	R/W	DELCRC	0	**Delayed CRC** Determines the number of bytes, if any, of proprietary header information that exist in the front of the IEEE 802.3 frames. 1 Four bytes of proprietary header information (ignored by the CRC calculation function). 0 Zero bytes of proprietary header information.

Table 61: MAC Configuration Register 2 bit definition

Bits	Access	Mnemonic	Reset	Description
D02	R/W	HUGE	0	**Huge frame enable** When this bit is set to 1, frames of any length are transmitted and received.
D01	R/W	FLENC	0	**Frame length checking** When set to 1, both transmit and receive frame lengths are compared to the Length/Type field. If the Length/Type field represents a length, the check is performed. Mismatches are reported on the transmit/receive statistics vector.
D00	R/W	FULLD	0	**Full duplex** 1 The MAC operates in full-duplex mode. 0 The MAC operates in half-duplex mode. It is recommended that you limit transmit frames to 512 bytes when full-duplex operation is enabled.

Table 61: MAC Configuration Register 2 bit definition

Pad operation table

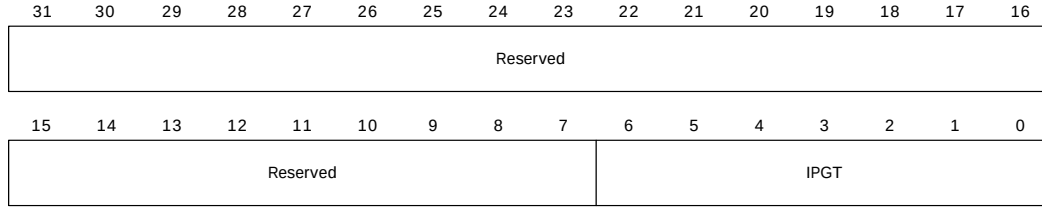
This table provides a description of the pad function based on the configuration of the AUTOP field.

Type	AUTOP	VLANP	PADEN	Action
Any	X	X	0	No pad; check CRC.
Any	0	0	0	Pad to 60 bytes; append CRC.
Any	X	1	1	Pad to 64 bytes; append CRC.
Any	1	0	1	■ If untagged, pad to 60 bytes and append CRC. ■ If VLAN-tagged, pad to 64 bytes and append CRC.

Table 62: PAD operation

Back-to-Back Inter-Packet-Gap register

Address: FF80 0408



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06:00	R/W	IPGT	N/A	**Back-to-back inter-packet-gap** A programmable field that represents the nibble time offset of the minimum possible period between the end of any transmitted packet to the beginning of the next packet. ■ **Full-duplex mode.** The register value should be the proper period in nibble times minus 3. The recommended setting is `'h15 (21d)`, which represents these minimum IPG values: — In 100 Mbps: 0.96 µs — In 10 Mbps: 9.6 µs ■ **Half-duplex mode.** The register value should be the proper period in nibble times minus 6. The recommended setting is `'h12 (18d)`, which represents these minimum IPG values: — In 100 Mbps: 0.96 µs — In 10 Mbps: 9.6 µs

Table 63: Back-to-Back Inter-Packet-Gap register bit definition

Non-Back-to-Back Inter-Packet-Gap register

Address: FF80 040C

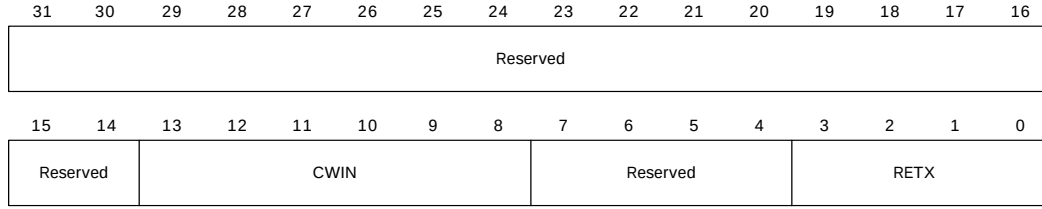
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:15	N/A	Reserved	N/A	N/A
D14:08	R/W	IPGR1	N/A	**Non back-to-back inter-packet-gap — part 1** A programmable field that represents the optional carrierSense window (referenced in IEEE 802.3 “Carrier Deference”). If carrier is found during IPGR1 timing, the MAC defers to the carrier. If carrier is found after IPGR1 times out, the MAC continues timing IPGR2. The transmitter causes a collision, activating the random back off fairness algorithm (ensuring fair access to the medium). The range of values for IPGR1 is `'h0–IPGR2`. The default value is `'hC (12d)`.
D07	N/A	Reserved	N/A	N/A
D06:00	R/W	IPGR2	N/A	**Non-back-to-back inter-packet-gap—part 2** A programmable field that represents the non-back-to-back inter-packet-gap. The default setting for IPGR2 is `'h12 (18d)`, which represents these minimum IPG values: ■ In 100 Mbps: 0.96 μs ■ In 10 Mbps: 9.6 μs

Table 64: Non-Back-to-Back Inter-Packet-Gap register bit definition

Collision Window/Collision Retry register

Address: FF80 0410



Register bit assignment

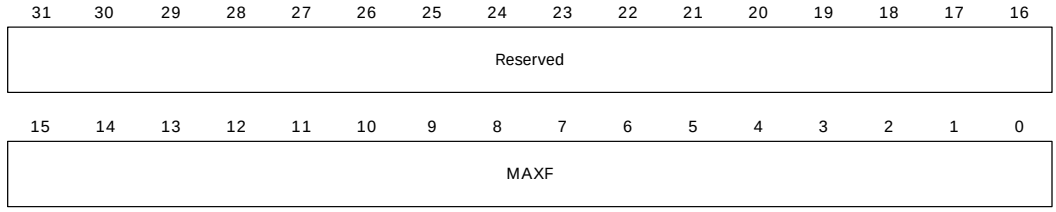
Bits	Access	Mnemonic	Reset	Description
D31:14	N/A	Reserved	N/A	N/A
D13:08	R/W	CWIN	'h37 (55d)	**Collision window** A programmable field that represents the slot time or collision window during which collisions occur in properly configured networks. The collision window starts at transmission, and includes the preamble and start frame delimiter (SFD). The default value for this field is 'h37 (55d), which corresponds to the frame byte count at the end of the window.
D07:04	N/A	Reserved	N/A	N/A
D03:00	R/W	RETX	'hF	**Retransmission maximum** A programmable field that specifies the number of retransmission attempts allowed after a collision, before aborting the packet due to excessive collisions. The 802.3u standard specifies an attemptLimit of 'hF (15d).

Table 65: Collision Window/Collision Retry register bit definition

Maximum Frame register

Address: FF80 0414

Register bit assignment

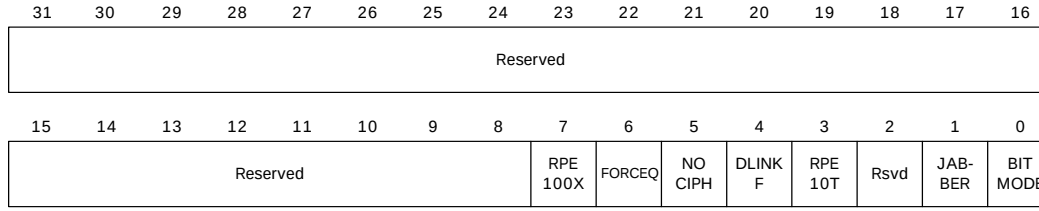


Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	MAXF	'h 0600	**Maximum frame length** Represents a maximum receive frame of 1536 octets (resets to 'h0600). An untagged maximum size Ethernet is 1518 octets. A tagged frame adds four octets, for a total of 1522 octets. Program this field if you want a shorter maximum length restriction. **Note:** If a proprietary header is allowed, adjust this field accordingly. For example, if 4-byte headers are appended to frames, the MAXF field should be set to 1526 octets. This setting allows the maximum VLAN tagged frame plus the 4-byte header.

Table 66: Maximum Frame register bit definition

PHY Support register

Address: FF80 0418



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07	R/W	RPE100X	0	**Reset PE100X module** Set this bit to 1 to reset the PE100X module, which contains the 4B/5B symbol encipher/ decipher logic. Must be set if not using 4B/5B logic.
D06	R/W	FORCEQ	0	**Force quiet** 1 Transmit data is quieted, which allows the cipher contents to be output. 0 (clear) Normal operation is enabled.
D05	R/W	NOCIPH	0	**No cipher** 1 (enable) The raw transmit 5B symbols are transmitted without ciphering. 0 (disable) Normal ciphering occurs.
D04	R/W	DLINKF	0	**Disable link fail** 1 Disables the 330 ms link fail timer, allowing for shorter simulations. 0 (clear) Normal operation occurs.
D03	R/W	RPE10T	0	**Reset PE10T module** Set this bit to 1 to reset the PE10T module. This module converts the MII nibble-streams to the serial bit stream for ENDEC transceivers. Must be set if not using the PE10T module.

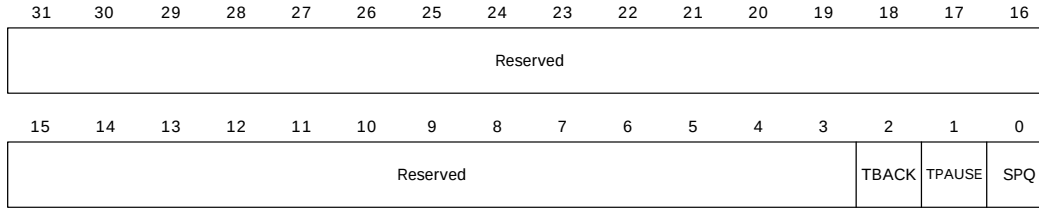
Table 67: PHY Support register bit assignment

Bits	Access	Mnemonic	Reset	Description
D02	N/A	Reserved	N/A	N/A
D01	R/W	JABBER	0	**Enable Jabber protection** *Jabber* is the condition in which a transmitter is stuck on longer than 50 ms, preventing other stations from transmitting. Set this bit to 1 to enable Jabber protection logic within the PE10T in ENDEC mode.
D00	R/W	BITMODE	0	**Bit mode** Set this bit to 1 to configure the MAC to operate in ENDEC mode. ENDEC mode is based on the bit-clock rather than the nibble-clock, and changes decodes (such as Excess Defer) to operate accordingly.

Table 67: PHY Support register bit assignment

Test register

Address: FF80 041C



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:03	N/A	Reserved	N/A	N/A

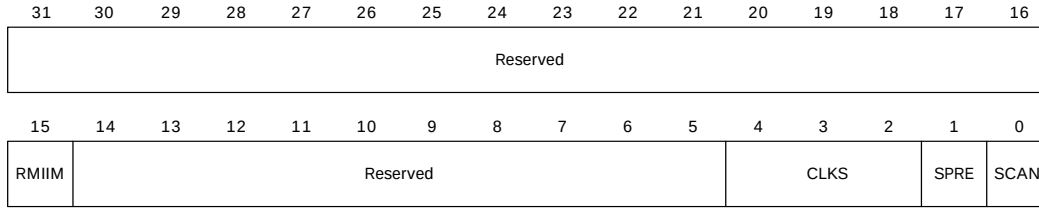
Table 68: Test register bit definition

Bits	Access	Mnemonic	Reset	Description
D02	R/W	TBACK	0	**Test backpressure** Set this bit to 1 to have the MAC assert backpressure on the link; this allows preamble to be transmitted raising carrier sense. A transmit packet from the system is sent during backpressure.
D01	R/W	TPAUSE	0	**Test pause** Set this bit to 1 to allow the MAC Control sublayer to inhibit transmissions, similar to receipt of a PAUSE receive control frame with a non-zero pause time parameter.
D00	R/W	SPQ	0	**Shortcut pause quanta** Set this bit to 1 to reduce the effective PAUSE quanta from 64 byte-times to 1 byte-time.

Table 68: Test register bit definition

MII Management Configuration register

Address: FF80 0420



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	RMIIM	0	**Reset MII management** Set this bit to 1 to reset the MII management module.
D14:05	N/A	Reserved	N/A	N/A
D04:02	R/W	CLKS	N/A	**Clock select** Used by the clock divide logic when creating the MII management clock (MDC pin). The IEEE 802.3u standard requires that the clock be no faster than 2.5 MHz. See Table 70: "CLKS field settings" on page 192 for examples. **Note:** Some PHY devices support clock rates up to 12.5 MHz.
D01	R/W	SPRE	0	**Suppress preamble** 1 The MII management module performs read/write cycles without the 32-bit preamble field. 0 Normal cycles are performed. **Note:** Some PHY devices support suppressed preamble.

Table 69: MII Management Configuration register bit definition

Bits	Access	Mnemonic	Reset	Description
D00	R/W	SCANI	0	**Scan increment (single scan for read data)** When set to 1, the MII management module performs read cycles across a range of PHY devices, from address 1 through the value set in the RADR field in the MII Address register (see page 194).

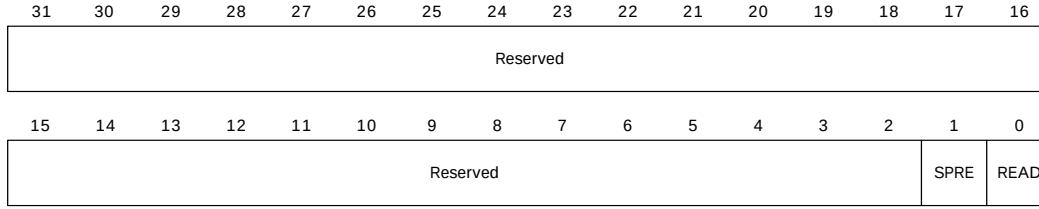
Table 69: MII Management Configuration register bit definition

CLKS field	SYS_CLK ratio	75 MHz example	12 MHz example
000	SYS_CLK / 4		
001	SYS_CLK / 4		
010	SYS_CLK / 6		2.0 MHz
011	SYS_CLK / 8		
100	SYS_CLK / 10		
101	SYS_CLK / 14		
110	SYS_CLK / 20		
111	SYS_CLK / 28	2.4 MHz	

Table 70: CLKS field settings

MII Management Command register

Address: FF80 0424



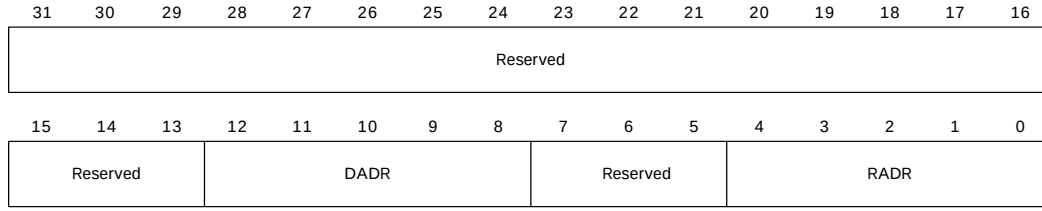
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:02	N/A	Reserved	N/A	N/A
D01	R/W	SCAN	0	**Automatically scan for read data** When this bit is set to 1, the MII management module performs read cycles continuously. This can be useful for monitoring link fail, for example.
D00	R/W	READ	0	**Single scan for read data** When this bit is set to 1, the MII management module performs a single read cycle. The read data is returned to the MII Read Data register (see page 196) after the BUSY bit in the MII Indicators register (see page 197) has returned a value of 0.

Table 71: MII Management Command register bit definition

MII Management Address register

Address: FF80 0428



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:13	N/A	Reserved	N/A	N/A
D12:08	R/W	DADR	0	**MII PHY device address** Represents the 5-bit PHY device address field for management cycles. Up to 31 different PHY devices can be addressed; address 0 is reserved.
D07:05	N/A	Reserved	N/A	N/A
D04:00	R/W	RADR	0	**MII PHY register address** Represents the 5-bit PHY register address field for management cycles. Up to 32 registers within a single PHY device can be addressed.

Table 72: MII Management Address register bit definition

MII Management Write Data register

Address: FF80 042C

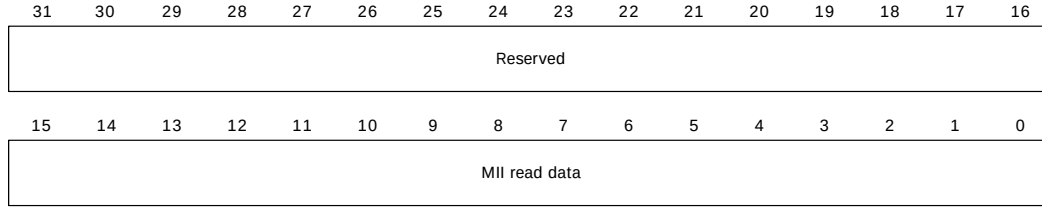
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	MWTD	N/A	**MII write data** When this register is written, an MII management write cycle is performed using the 16-bit data defined in the PHY Address register by the preconfigured PHY device and PHY register addresses. The write operation is completed when the BUSY bit in the MII Indicators register (see page 197) returns to 0.

Table 73: MII Management Write Data register bit definition

MII Management Read Data register

Address: FF80 0430



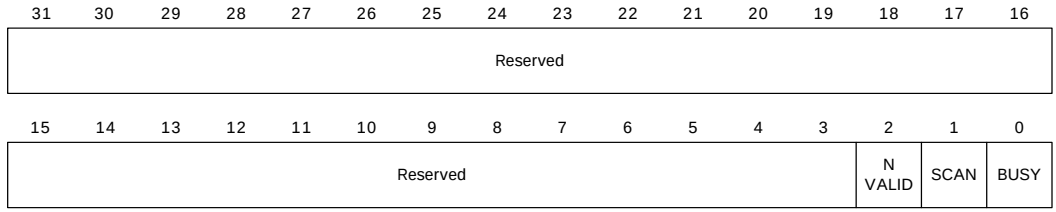
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R	MRDD	N/A	**MII read data** Provides read data following an MII management read cycle. An MII management read cycle is executed by loading the PHY Address register, then setting the READ bit in the MII Indicators register (see page 197) to 1. Read data is available after the BUSY bit in the MII Indicators register returns to 0.

Table 74: MII Management Read Data register bit definition

MII Management Indicators register

Address: FF80 0434



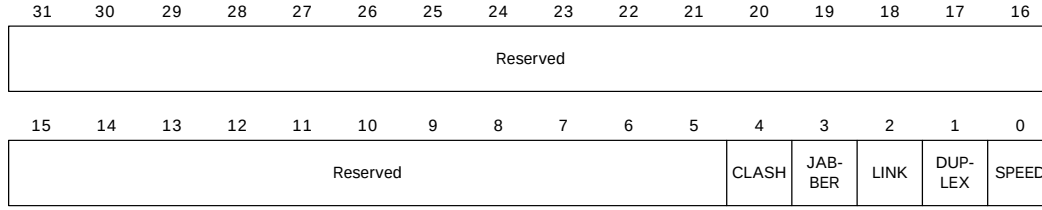
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:03	N/A	Reserved	N/A	N/A
D02	R	NVALID	N/A	**Read data not valid** A 1 indicates that the MII management read cycle is not complete and read data is not yet valid.
D01	R	SCAN	N/A	**Automatically scan for read data in progress** A 1 indicates that continuous MII management scanning read operations are in progress.
D00	R	BUSY	N/A	**MII interface BUSY with read/write operation** A 1 indicates that the MII management module is currently performing an MII management read or write cycle. This bit returns to 0 when the operation completes.

Table 75: MII Management Indicators register bit definition

SMII Status register

Address: FF80 0438



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:05	N/A	Reserved	N/A	N/A
D04	R	CLASH	N/A	**MAC-to-MAC with PHY** A 1 indicates that MAC-to-MAC mode is selected, but a PHY is detected.
D03	R	JABBER	N/A	**Jabber condition present** A 1 indicates that a Jabber condition is present.
D02	R	LINK	N/A	**Link OK** A 1 indicates LINK OK.
D01	R	DUPLEX	N/A	**Full-duplex operation** 1 Indicates full-duplex operation 0 Indicates half-duplex operation
D00	R	SPEED	N/A	**100 Mbps** 1 Indicates 100 Mbps 0 Indicates 10 Mbps

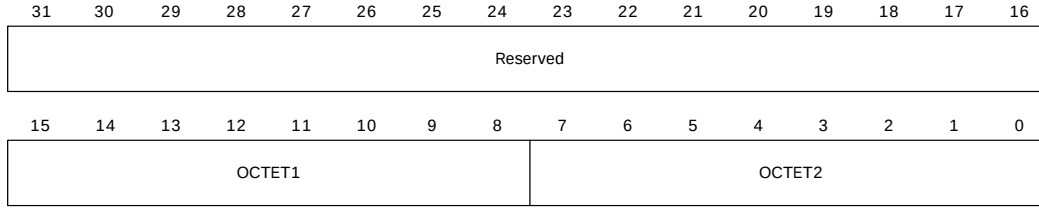
Table 76: SMII Status register bit definition

Station Address registers

The station address is used for address comparison of inbound receive packets. The 48-bit address value is held in three registers, each containing 16 bits.

Station Address Register 1

Address: FF80 0440

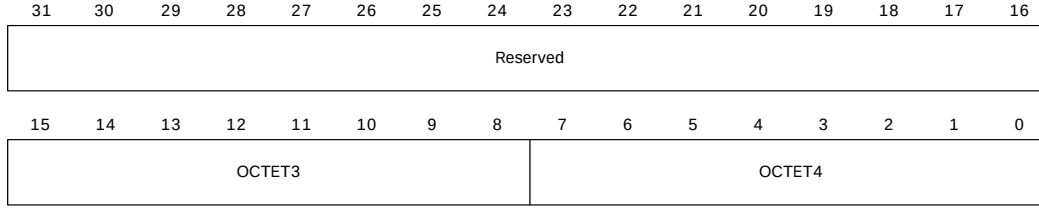


Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:08	R/W	OCTET1	'h00	**Station address octet 1** Holds the first octet of the station address (bits 47:40).
D07:00	R/W	OCTET2	'h00	**Station address octet 2** Holds the second octet of the station address (bits 39:32).

Table 77: Station Address Register 1 bit definition

Station Address Register 2

Address: FF80 0444



Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A

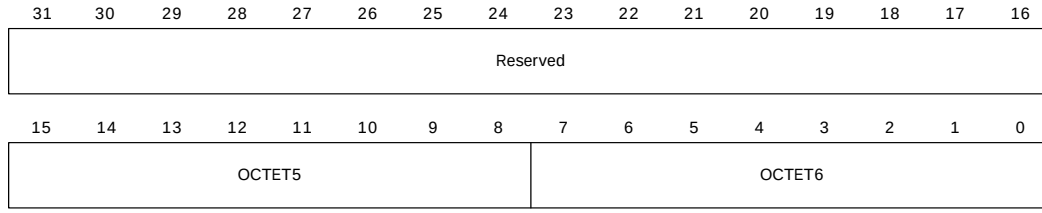
Table 78: Station Address Register 2 bit definition

Bits	Access	Mnemonic	Reset	Description
D15:08	R/W	OCTET3	'h00	**Station address octet 3** Holds the third octet of the station address (bits 31:24).
D07:00	R/W	OCTET4	'h00	**Station address octet 4** Holds the fourth octet of the station address (bits 23:16).

Table 78: Station Address Register 2 bit definition

Station Address Register 3

Address: FF80 0448

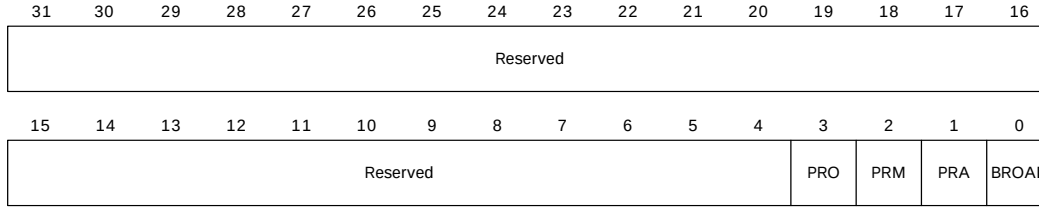


Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:08	R/W	OCTET5	'h00	**Station address octet 5** Holds the fifth octet of the station address (bits 15:08).
D07:00	R/W	OCTET6	'h00	**Station address octet 6** Holds the sixth octet of the station address (bits 07:00).

Table 79: Station Address Register 3 bit definition

Station Address Filter register

Address: FF80 05C0



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R/W	PRO	0	**Enable promiscuous mode (receive *all* packets)** A 1 indicates promiscuous mode. The MAC receives all packets, without any filtering.
D02	R/W	PRM	0	**Accept ALL multicast packets** A 1 allows the MAC to receive all multicast Ethernet packets, without any filtering.
D01	R/W	PRA	0	**Accept multicast packets using hash table** 1 allows the MAC to receive only those multicast Ethernet packets that are selected by the multicast hash table (see "Multicast hash table entries and bit definitions" on page 202).
D00	R/W	BROAD	0	**Accept ALL broadcast packets** A 1 allows the MAC to receive all broadcast Ethernet packets, without any filtering.

Table 80: Station Address Filter register bit definition

Register hash table

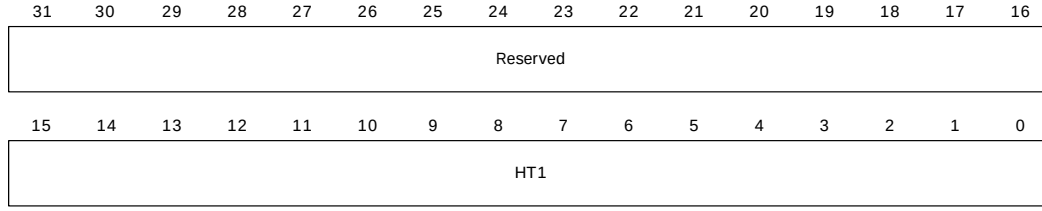
The MAC receiver provides the station address logic (SAL) with a 6-bit CRC value. This value points to a bit position in the 64-bit multicast hash table. A receive packet is accepted if the current frame is a multicast frame and the 6-bit CRC addresses a bit in the hash table that is set to 1. Otherwise, the packet is rejected.

The 64-bit address value is held in four registers, each containing 16 bits. The least significant bit of the first register corresponds to a CRC value of 0. The most significant bit of the last register corresponds to the CRC value of 63.

The multicast address bytes are loaded into the Station Address registers in Little Endian format.

Multicast hash table entries and bit definitions

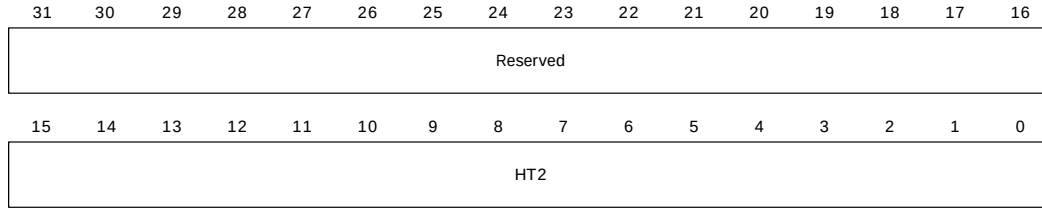
Address: FF80 05D0



Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	HT1	0	CRC value 15–0

Table 81: HT1 bit definition

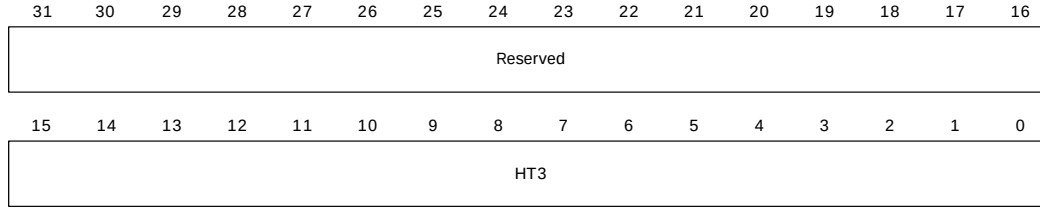
Address: FF80 05D4



Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	HT2	0	CRC values 31–16

Table 82: HT2 bit definition

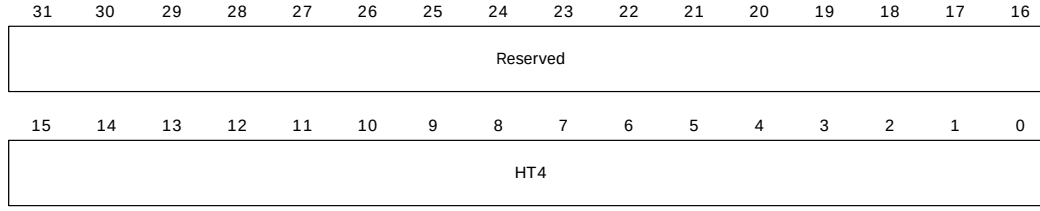
Address: FF80 05D8



Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	HT2	0	CRC values 47–32

Table 83: HT3 bit definition

Address: FF80 05DC



Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	HT4	0	CRC values 63–48

Table 84: HT4 bit definition

Calculating hash table entries

The C code in this section describes how to calculate the hash table entries based on 6-byte Ethernet destination addresses.

```
static ETH_ADDRESS mca_address[MAX_MCA];      /*list of MCA addresses*/
static INT16 mca_count;                       /*# of MCA addresses*/
/*
 *
 *   Function: void eth_load_mca_table (void)
 *
 *   Description:
 *
 *   This routine loads the MCA table. It generates a hash table for
 *   the MCA addresses currently registered and then loads this table
 *   into the NS7520 chip.
 *
 *   Parameters:
 *
 *      none
 *
 *   Return values
 *
 *      none
 *
 */

static void eth_load_mca_table (void)

{
     WORD16 hash_table[4];

     // create hash table for MAC address
     eth_make_hash_table (hash_table);

        (*NS7520_EFE) .ht4.bits.data = SWAP16(hash_table[3]);
        (*NS7520_EFE) .ht3.bits.data = SWAP16(hash_table[2]);
        (*NS7520_EFE) .ht2.bits.data = SWAP16(hash_table[1]);
        (*NS7520_EFE) .ht1.bits.data = SWAP16(hash_table[0]);
}

/*
 *
 *   Function: void eth_make_hash_table
 *
 *   Description:
```

```
 *
 *   This routine creates a hash table based on the CRC values of
 *   the MAC addresses set up by eth_add_mca(). The CRC value of
 *   each MAC address is calculated and the lower six bits are used
 *   to generate a value between 0 and 64. The corresponding bit in
 *   the 64-bit hash table is then set.
 *
 *   Parameters:
 *
 *      hash_table                  pointer to buffer to store hash table in
 *
 *   Return Values:
 *
 *      none
 *
 */

static void eth_make_has_table (WORD16 *hash_table)

{
        int index:

        memset (hash_table, 0, 8);                /* clear hash table*/

        for (index = 0; index < mca_count; index++)/* for each mca
address*/
        {
           set_has_bit ((BYTE *) hash_table, calculate_hash_bit
mca_address [index]));
        }
}

/*
 *
 *   Function: void set_hash_bit (BYTE *table, int bit)
 *
 *   Description:
 *
 *      This routine sets the appropriate bit in the hash table.
 *
 *   Parameters:
 *
 *      table   pointer to hash table
 *      bit     position of bit to set
 *
 *   Return Values:
```

```
 *
 *      none
 *
 */

static void set_hash_bit (BYTE *table, int bit)

{
        int byte_index, bit_index;

        byte_index = bit >> 3;
        bit_index = bit & 7;
        table [byte_index] |= (1 << bit_index);
}

/*
 *
 *   Function: int calculate_hash_bit (BYTE *mca)
 *
 *   Description:
 *
 *      This routine calculates which bit in the CRC hash table needs
 *      to be set for the NET+20UM to recognize incoming packets with the
 *      MCA passed to us.
 *
 *   Parameters:
 *
 *      mca     pointer to multi-cast address
 *
 *   Return Values:
 *
 *      bit position to set in hash table
 *
 */

#define POLYNOMIAL 0x4c11db6L

static int calculate_hash_bit (BYTE *mca)

{
        WORD32 crc;
        WORD16 *mcap, bp, bx;
        int result, index, mca_word, bit_index;
        BYTE lsb;
        WORD16 copy_mca[3];
```

```
        memcpy (copy_mca, mca, sizeof (copy_mca));
        for (index = 0; index < 3’ index++)
        {
           copy_mca [index] = SWAP16 (copy_mca [index]);
        }

        mcap = copy mca;
        crc = 0xffffffffL;

        for (mca_word = 0; mca_word < 3; mca_word++)
        {
           bp = *mcap;
           mcap++;
           for (bit_index = 0; bit_index < 16; bit_index++)
           {
                bx = (WORD16) (crc >> 16);     /* get high word of crc*/
                bx = rotate (bx, LEFT, 1);     /* bit 31 to lsb*/
                bx ^+ bp;                      /* combine with incoming*/
                crc <<=1;                      /* shift crc left 1 bit*/
                bx &= 1;                       /* get control bit*/
                if (bx)                        /* if bit set*/
                {
                   crc ^= POLYNOMIAL;          /* xero crc with polynomial*/
                }
                crc |= bx;                     /* or in control bit*/
           }
        }

     // CRC calculation done. The 6-bit result resides in bit
     // locations 28:23

     result = (crc >> 23) & 0x3f;

     return result;

}
```

Serial Controller Module

CHAPTER 10

The NS7520 supports two independent universal asynchronous/synchronous receiver/transmitter channels. Each channel supports several modes, conditions, and formats.

Supported features

Each serial channel supports these features:

- Independent programmable bit-rate generator
- UART and SPI modes
- High speed data transfer
  - x1 mode: 4 Mbps
  - x16 mode: 230 Kbits/sec
- 32-byte TX FIFO
- 32-byte RX FIFO
- Programmable data format
  - 5 to 8 data bits
  - Odd, even, or no parity
  - 1,2 stop bits
- Programmable channel modes
  - Normal
  - Local loopback
  - Remote loopback
- Control signal support
  - RTS, CTS, DTR, DSR, DCD, RI
- Maskable interrupt conditions
  - Receive break detection
  - Receive framing error
  - Receive parity error
  - Receive overrun error
  - Receive FIFO ready
  - Receive FIFO half-full
  - Transmit FIFO ready
  - Transmit FIFO half-empty

  - CTS, DSR, DCD, RI state change detection
- Clock/Data encoding
  - NRZ
  - NRZB
  - NRZI
  - FM
  - Manchester
- Multi-drop capable

Figure 24 shows the structure of the serial module.

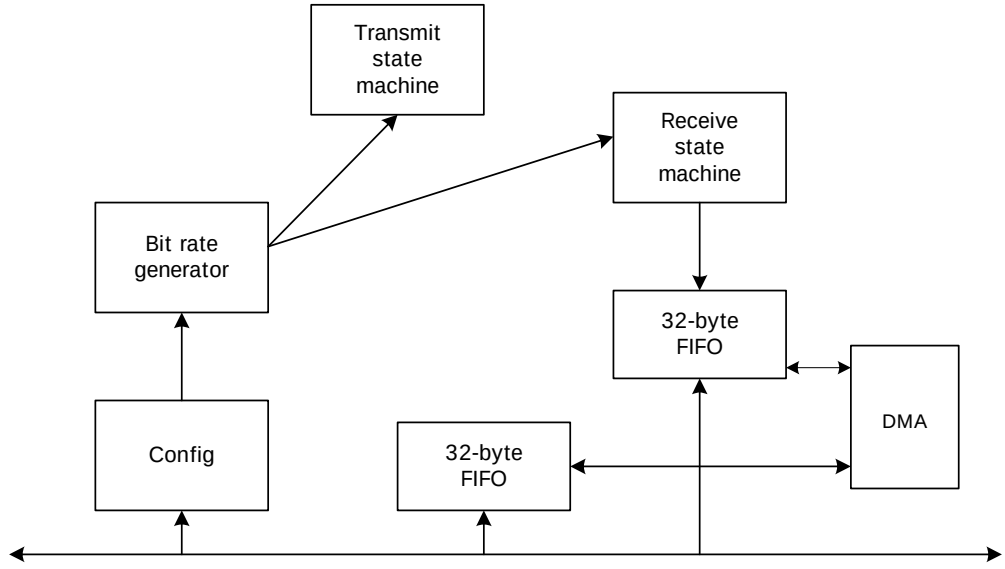


Figure 24: Serial port block diagram

Bit-rate generator

Each serial channel supports an independent programmable bit-rate generator. The bit-rate generator runs both the transmitter and receiver of a given channel (no split speed support).

You can configure the bit-rate generator to use the external oscillator, the external clock input, or internal system timing as its timing reference. This allows for a wider range of possible bit-rates. See "Serial Channel 1, 2 Bit-Rate registers" on page 244 for more information.

Serial protocols

The serial port provides support for these protocols:

- UART (Universal Asynchronous Receiver Transmitter)
- SPI

UART mode

Many applications require a simple mechanism for sending low-speed information between two pieces of equipment. The universal asynchronous receiver transmitter (UART) protocol is the de facto standard for simple serial communications.

The UART receiver uses an over-sampling technique to find the bit-level framing of the UART protocol. The framing protocol is as follows:

```
Start bit:      0
Data:           5, 6, 7, or 8 bits
Parity:         Odd, even, or no parity
Code:           1 or more
```

Because the transmitter and receiver operate asynchronously, the transmit and receive clocks between them do not need to be connected. Instead, the receiver over-samples the receive data stream by a factor of 8, 16, or 32. A byte of data is preceded by a high-to-low transition at the beginning of the start bit. The receiver continuously samples the data stream to determine the center of each bit-time.

When the UART is not transmitting data, it transmits a continuous stream of ones — considered the *idle* condition. When data transmission begins again, the transmitter sends the start bit and the receiver is enabled.

You can configure the UART to perform the following functions:

- **Enable the transmitter using the CTS handshaking signal.** In this mode, the transmitter cannot start a new UART data frame unless CTS is active. If CTS is dropped anywhere in the middle of a UART data frame, the current character is completed but the next character is stalled.
- **Signal its receiver FIFO status using the RTS handshaking signal.** When the receive FIFO has only four characters of available space, the RTS signal is dropped. The RTS and CTS pairs can be used for hardware flow control.
- **Operate in synchronous timing mode.** When using synchronous timing mode, a clock is provided with each bit; an over-sampling technique is not required. Transmit and receive clocks must be connected between two end points.

See "General-purpose I/O configurations" on page 222 and "Serial Channel registers" on page 225 for information about signals and configuration.

SPI mode

The SPI (serial peripheral interface) controller provides:

- **Full-duplex, synchronous interface.** The master interface operates in a broadcast mode, activating the slave interface with the select (SEL_) signal. The master interface can also be configured to address more than one slave interface using GPIO pins.
- **Four-wire connection (TXD, RXD, CLK, SEL_).** The transmitter and receiver use the same clock. When configured in master mode, the channel's bit-rate generator provides the timing reference.
- **Character-oriented data channel.** The protocol can provide simple parallel/serial data conversion to stream serial data between memory and a peripheral.
- **Master and slave configurations.** The SPI port is capable of simultaneous full-duplex operation. The transfer of information is controlled by a single clock signal.
  - For the SPI master interface, the clock signal is an output.
  - For the SPI slave interface, the clock signal is an input.

Information transfer is also qualified with an enable signal. The SPI enable signal must be active low for data transfer to occur, regardless of the SPI clock signal. The SPI enable function allows multiple slaves to be individually addressed in a multi-drop configuration.

Note: The NS7520 supports only SPI modes 0 and 1. See Figure 25, "SPI master mode 0 and 1 two-byte transfer," on page 219 and Figure 26, "SPI slave mode 0 and 1 two-byte transfer," on page 222 for illustrations.

FIFO management

Data flow between the SPI master/slave interfaces and memory occurs through the FIFO blocks within the SER module. Each serial port provides a 32-byte transmit FIFO and a 32-byte receive FIFO. Both the transmit and receive FIFOs are memory-mapped to the processor address space.

Transmit FIFO interface

The processor can write either 1, 2, or 4 bytes at a time to the transmit FIFO. The number of bytes written is controlled by the data size defined by the ARM processor as shown in these examples:

Terminology	What's being written	Value
C	Byte using a byte pointer	`(char*)0xffd00010=(char)data`
C	2 bytes using a short pointer	`(short)0xffd00010=(short)data`
C	4 bytes using a standard long word pointer	`(long)0xffd00010=(long)data`
Assembler	Byte	STRB instructions
Assembler	Half word	STRH instructions
Assembler	Long word	STR instructions

Operating in Endian modes

- **Big Endian mode configuration.** Transmits first the most significant bytes in the word written to the FIFO. For example, the long word `0x11223344` results in the character `0x11` being transmitted first and `0x44` being transmitted last.

- **Little Endian mode configuration.** Transmits first the least significant bytes in the word written to the FIFO. For example, the long word `0x11223344` results in the character `0x44` being transmitted first and `0x11` being transmitted last.

Processor interrupts vs. DMA
The transmit FIFO can be filled using processor interrupts or DMA.

- When using processor interrupts, the processor can write one long word (4 bytes) of data to the transmit FIFO when the TRDY bit in Serial Channel Status Register A is active high. If the THALF bit in Serial Channel Status Register A is active high, the processor can write four long words (16 bytes) of data to the transmit FIFO. To facilitate an interrupt when either the TRDY or THALF status bit is active, the processor can set one or both of the corresponding interrupt enables — ETXRDY, ETXHALF — in Serial Channel Control Register A. The appropriate interrupt enable bit — SER 1 TX or SER 2 TX — in the GEN module Interrupt Enable register must also be set.
- When using the DMA controller, the processor must interface with the DMA channel registers and the DMA buffer descriptor block attached to DMA channels 8 and 10. To facilitate the use of transmit DMA, the ETXDMA bit in Serial Channel Control Register A must be set active high and the serial transmitter interrupts should be disabled.

Receive FIFO interface

The receive FIFO presents up to four bytes of data at a time to the processor interface. The number of valid bytes found in the next read of the FIFO is defined by the information found in the RXFDB field in Serial Channel Status Register A.

The Endian rules described for the transmit FIFO also apply for the receive FIFO; see "Operating in Endian modes" on page 214.

When reading from the receive FIFO, the processor must perform a long word read operation. Each time a read cycle is performed, the receive FIFO goes to the next long word entry. The processor cannot read individual bytes from the same FIFO long word entry.

Processor interrupts vs. DMA
The receive FIFO can be emptied using processor interrupts or DMA.

Using processor interrupts

When using processor interrupts, the processor can read one long word (4 bytes) of data from the receive FIFO when the RRDY bit in Serial Channel Status Register A is active high. This long word can have 1, 2, 3, or 4 bytes of valid data within the word. The number of valid bytes is determined by the bit encoding in the RXFDB field in Serial Channel Status Register A. The RXFDB field must be read before the FIFO Data register is read.

The RBC (receive buffer closed) bit in Serial Channel Status register A indicates that a receive data buffer has been closed and receiver status can be read from that register. Before additional data can be read from the FIFO, the RBC bit must be acknowledged by writing a 1 to the same bit position in the Serial Channel Status Register A.

This is the recommended process flow for the serial port receiver interrupt service routine:

1. Read Serial Channel Status Register A.
2. If RBC is true:
   - **a** Record receiver buffer closed status (if desired).
   - **b** Write a 1 to the RBC bit position in Serial Channel Status Register A.
   - **c** Read Serial Channel Status Register A again.
3. If RRDY is true:
   - **a** Read the data FIFO.
   - **b** Use the RXFDB field to pick valid bytes.

   To facilitate an interrupt when either the RRDY or RBC status bits are active, the processor must set one or both of the corresponding interrupt enables — ERXDRDY, ERXBC — in Serial Channel Control Register A. The appropriate interrupt enable bit — SER 1 RX, SER 2 RX — in the GEN module Interrupt Enable register must also be set.

Using DMA

When using the DMA controller, the processor must interface with the DMA channel registers and the DMA buffer descriptor block attached to DMA channels 7 and 9. To facilitate the use of receive DMA, the ERXDMA bit in Serial Channel Control Register A must be set active high and the serial receiver interrupts should be disabled.

SPI master mode

SPI master mode controls the flow of data between memory in the master SPI interface and an external SPI slave peripheral. The SPI master determines the number of bytes for transfer. The SPI master port simultaneously transmits and receives the same number of bytes. A single clock signal controls the transfer of information; for SPI master mode, the signal is an output. Information is also qualified with an enable signal, which is controlled by the SPI master. The SPI enable signal must be active low for data transfer to occur, regardless of the SPI clock signal. The SPI enable function allows for multiple slaves to be addressed individually in a multi-drop configuration.

Signals

The GEN module must be configured appropriately to allow the SPI interface signals to interface with the PORTA and PORTC GPIO pins (see "PORTA Configuration register" on page 74 and "PORTC Configuration register" on page 77).

Configuration

The SER module must be configured properly to operate in either master or slave mode. For master mode operation, the MODE field in Serial Channel Control Register B must be set to 10 before the CE field in Serial Channel Control Register A is set to 1.

Use this suggested configuration order for SPI master mode:

1. Reset the serial port by writing a 0 to Serial Channel Control Register A.
2. Configure the Serial Channel Bit-Rate register as shown:
   - EBIT: 1 for enable
   - TMODE: 1 for 1x mode
   - RXSRC: 0 for internal
   - TXSRC: 0 for internal
   - RXEXT: 0 for disable
   - TXEXT: either 1 or 0 — the setting has no effect
   - CLKMUX: user-defined
   - TXCINV: 0 for normal
   - RXCINV: 0 for normal
   - N: user–defined

3 Configure the buffer GAP timer, if you want. The buffer GAP timer terminates a DMA transfer at a programmable interval from the time the first character is received. (See "Serial Channel 1, 2 Receive Buffer Gap Timer," beginning on page 255, for more information).

4 Configure the character GAP timer, if you want. The character GAP timer terminates a DMA transfer if the time between the receipt of two characters exceeds a programmable interval. (See "Serial Channel 1, 2 Receive Character Gap Timer," beginning on page 256, for more information.)

5 Configure Serial Channel Control Register B as shown:
   - RBGT: 1 to enable the buffer GAP timer
   - RCGT: 1 to enable the character GAP timer
   - MODE: 10 for master mode
   - BITORDR: user-defined

6 Configure Serial Channel Control Register A as shown:
   - CE: 1 for enable
   - WLS: 11 for 8-bit operation

SPI master transmitter

The SPI master transmitter operates as follows:

- Changes its TXD output on the falling edge of the SPI clock signal while the SPI enable signal is driven active low. The SPI slave devices should sample data in the rising edge of the SPI clock signal.
- Drives the SPI enable signal active low from the falling edge of the SPI clock for the first bit of a byte being transmitted, and inactive high after the rising edge of the SPI clock signal during the eighth bit of the byte currently transmitted.
- Drives the SPI enable signal active low to identify when data is being transmitted. The SPI clock signal never transitions from low to high while the internal SPI enable signal is inactive high.
- Transmits bytes when data is available in the TX FIFO. When the TX FIFO becomes empty, the SPI enable signal is driven inactive high until more data is available in the TX FIFO.

SPI master receiver

The SPI master receiver operates as follows:

- Samples the RXD input on the rising edge of the SPI clock signal while the SPI enable signal is driven active low.
- Receives one byte of inbound data for each byte of transmit data sent. The SPI master receiver cannot receive more data that what is transmitted.

  When the SPI master receiver collects four bytes, those four bytes are written to the RX FIFO. Receive data is read by the CPU or DMA controller from the other side of the FIFO. If the SPI master transmitter always sends data in multiples of four bytes, the SPI master receiver operates smoothly without any restrictions.

  When the SPI master transmitter sends an odd number of bytes, the SPI master receiver needs to wait for the fourth byte before insertion into the FIFO. This can result in stale data sitting in the SPI master receiver. To commit these residual bytes to the RX FIFO, the buffer and/or character GAP timers must be used. When either timer expires, any residual RX data bytes are immediately written to the RX FIFO. Note that some delay will occur in writing the final residual bytes; the delay is determined by the configuration of the buffer and character GAP timers.

Figure 25 shows a two-byte transfer in SPI master modes 0 and 1. See "Serial internal/external timing" on page 303 for associated timing values.

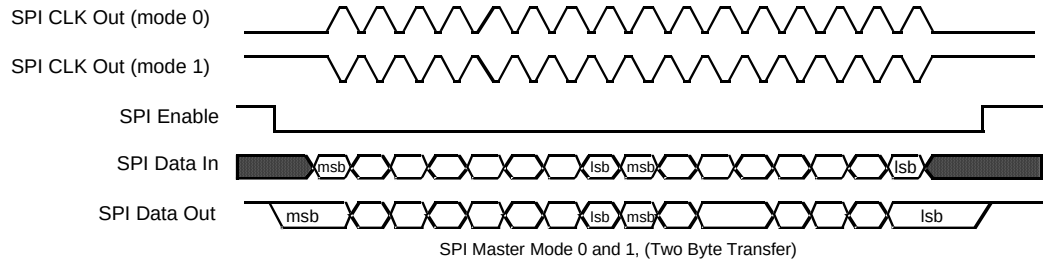


Figure 25: SPI master mode 0 and 1 two-byte transfer

SPI slave mode

SPI slave mode supports the peripheral side of an SPI interface. The SPI master controls the number of bytes for the transfer. The SPI slave port simultaneously transmits and receives the same number of bytes. The transfer of information is

controlled by a single clock signal; for SPI slave mode, the signal is an input. Information transfer is also qualified with an enable signal input, which is controlled by the SPI master. The SPI enable signal must be active low for data transfer to occur, regardless of the SPI clock signal. The SPI enable function allows for multiple slaves to be addressed individually during a multi-drop configuration.

Signals

The GEN module must be configured appropriately to allow the SPI interface signals to interface with the PORTA and PORTC GPIO pins (see "PORTA Configuration register" on page 74 and "PORTC Configuration register" on page 77).

Configuration

The SER module must be configured properly to operate in either master or slave mode. For slave mode operation, the MODE field in Serial Channel Control Register B must be set to 11 before the CE field in Serial Channel Control Register A is set to 1.

Use this suggested configuration order for SPI slave mode:

1. Reset the serial port by writing a 0 to Serial Channel Control Register a.
2. Configure the Serial Channel Bit-Rate register, as shown:
   - EBIT: 1 for enable
   - TMODE: 1 for 1x mode
   - RXSRC: 1 for external;
   - TXSRC: 1 for external;
   - RXEXT: 0 for disable
   - TXEXT: 0 for disable
   - CLKMUX: n/a (external timing source)
   - TXCINV: 0 for normal
   - RXCINV: 0 for normal
   - N: n/a (external timing source)
3. Configure the buffer GAP timer, if you want. The buffer GAP timer terminates a DMA transfer at a programmable interval from the time the first character is received. (See "Serial Channel 1, 2 Receive Buffer Gap Timer," beginning on page 255, for more information).
4. Configure the character GAP timer, if you want. The character GAP timer terminates a DMA transfer if the time between the receipt of two characters

exceeds a programmable interval. (See "Serial Channel 1, 2 Receive Character Gap Timer," beginning on page 256, for more information.)

5 Configure Serial Channel Control Register B, as shown:

- RBGT: 1 to enable the buffer GAP timer
- RCGT: 1 to enable the character GAP timer
- MODE: 11 for slave mode
- BITORDR: user-defined

6 Configure Serial Channel Control Register A, as shown:

- CE: 1 for enable
- WLS: 11 for 8-bit operation

SPI slave transmitter

The SPI slave transmitter operates as follows:

- Has the first bit ready for transmission before the SPI enable signal is activated by the SPI master.
- Changes the TXD output to the next data bit for transmission on the rising edge of the SPI clock, while the SPI enable signal is active low.
- Goes through a sampling of the SPI clock input. The output changes 3 to 4 internal system clock ticks from the rising edge of the SPI clock input. The SPI master receiver does not sample the changing data until the next rising edge of SPI clock.
- Continues to change TXD data bits on the rising edge of SPI clock until the SPI enable signal is driven inactive high. While the SPI enable signal is inactive high, the TXD output remains constant.

SPI slave receiver

The SPI slave receiver operates as follows:

- Samples the RXD input on the rising edge of the SPI clock signal while the SPI enable signal is driven active low. The SPI slave receiver actually goes through a sampling of the SPI input, which means the input is sampled 3 to 4 internal system clock ticks from the rising edge of the SPI clock input.
- When the SPI slave receiver collects four bytes, these four bytes are written to the RX FIFO. Receive data is read by the CPU or DMA controller from the other side of the FIFO. If the SPI master transmitter always sends data in

multiples of four bytes, the SPI slave receiver operates smoothly without any restrictions.

When the master SPI transmitter sends an odd number of bytes, the SPI slave receiver waits for the fourth byte before insertion into the FIFO. This can result in stale data sitting in the SPI slave receiver. To commit these residual bytes to the RX FIFO, the buffer and/or character GAP timers must be used. When either timer expires, any residual RX data bytes are immediately written to the RX FIFO. Note that some delay will occur in writing the final residual bytes; the delay is determined by the configuration of the buffer and character GAP timers.

Figure 26 shows a two-byte transfer in SPI slave modes 0 and 1. See "Serial internal/external timing" on page 303 for associated timing values.

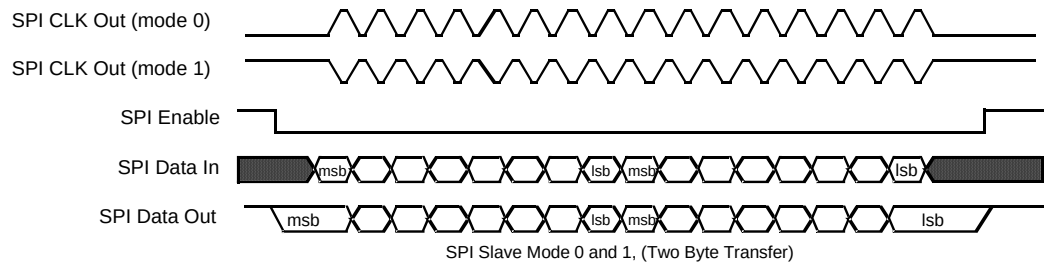


Figure 26: SPI slave mode 0 and 1 two-byte transfer

General-purpose I/O configurations

The GEN module provides the physical layer connections for the NMSI (Non Multiplexed Serial Interface) and SPI interfaces. NMSI is used for the UART protocol. Use the appropriate GPIO port pins to correspond with the NMSI signals (see "PORTA Configuration register" on page 74 and "PORTC Configuration register" on page 77).

When NMSI is used with synchronous operation, the RXCA or B and TXCA or B signals provide the RX and TX clocks, respectively. When NMSI is used with asynchronous operation, the OUT1A or B and OUT2A or B signals provide the general-purpose output pin. See "Serial Channel 1, 2 Bit-Rate registers," beginning on page 244, for more information about this synchronous/asynchronous setting.

Serial port performance

The serial ports have a finite performance limit on their ability to handle various serial protocols. Performance is limited by the speed of the SYSCLK operating the NS7520. The configured speed for the internal PLL defines the SYSCLK rate, as shown in this table:

Operating Mode	Serial Port Maximum Rate
UART (x16)	SYSCLK/64
UART (x1)	SYSCLK/4
SPI	SYSCLK/8

Configuration

The serial controller module has a block of configuration space as shown:

Address	Register
FFD0 0000	Channel 1 Control Register A
FFD0 0004	Channel 1 Control Register B
FFD0 0008	Channel 1 Status Register A
FFD0 000C	Channel 1 Bit-Rate register
FFD0 0010	Channel 1 FIFO Data register
FFD0 0014	Channel 1 Receive Buffer Gap Timer
FFD0 0018	Channel 1 Receive Character Gap Timer
FFD0 001C	Channel 1 Receive Match register
FFD0 0020	Channel 1 Receive Match Mask register
FFD0 0040	Channel 2 Control Register A
FFD0 0044	Channel 2 Control Register B
FFD0 0048	Channel 2 Status Register A

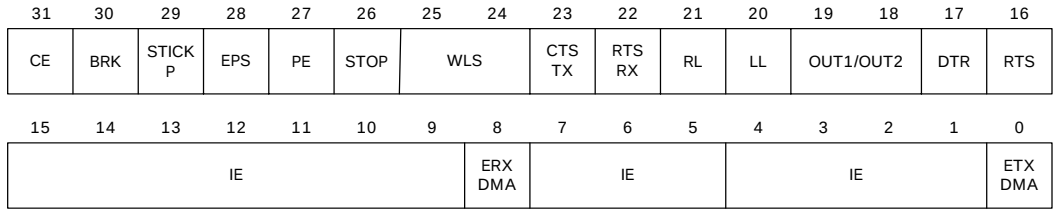
Address	Register
FFD0 004C	Channel 2 Bit-Rate register
FFD0 0050	Channel 2 FIFO Data register
FFD0 0054	Channel 2 Receive Buffer Gap Timer
FFD0 0058	Channel 2 Receive Character Gap Timer
FFD0 005C	Channel 2 Receive Match register
FFD0 0060	Channel 2 Receive Match Mask register

Serial Channel registers

All control bits are active high unless followed by an underscore (_); the underscore indicates active low.

Serial Channel 1, 2 Control Register A

Address: FFD0 0000 / 40



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	CE	0	**Channel enable** 0 Resets the port and the data FIFOs (disables the serial channel) 1 Enables serial channel operation The CE field cannot be set until the MODE field is configured in Serial Channel Control Register B.
D30	R/W	BRK	0	**Send break** Forces a break condition in UART mode. While BRK is set to 1, the UART transmitter outputs a logic 0 or a space condition, on the TXD output signal.

Table 85: Serial Channel Control Register A

Bits	Access	Mnemonic	Reset	Description
D29	R/W	STICKP	0	**Stick parity** Can be used to force the UART parity field to a certain state, as defined by the EPS field, instead of a parity bit calculated against the data word. STICKP applies only when the PE field is also set to 1. Set the STICKP field to 1 to force transmission of the static parity value.
D28	R/W	EPS	0	**Even parity select** 0 Odd parity 1 Even parity Determines whether the serial channel uses odd or even parity when calculating the parity bit in UART mode. When the STICKP field is set, the EPS field defines the static state for the parity bit.
D27	R/W	PE	0	**Parity enable** When set, parity is enabled for the UAT transmitter and receiver. The transmitter generates proper parity. The receiver checks for proper parity. If the receiver encounters a bad parity bit, the RPE field is set in Serial Channel Status Register A.
D26	R/W	STOP	0	**Number of stop bits** 0 1 stop bit 1 1.5 stops for a 5-bit word, 2 stop bits for other words. Determines the number of stop bits in each UART transmitter.
D25:24	R/W	WLS	0	**Data word length select** 00 5 bits 01 6 bits 10 7 bits 11 8 bits Determines the number of data bits in each UART data word.

Table 85: Serial Channel Control Register A

Bits	Access	Mnemonic	Reset	Description
D23	R/W	CTSTX	0	**Enable the transmitter with active CTS** Supports hardware handshaking. When CTSTX is set, the transmitter operates only when the external CTS signal is in the active state. An external device can use CTS to temporarily stall data transmission.
D22	R/W	RTSRX	0	**Enable active RTS** (only when RX FIFO has space) Supports hardware handshaking. When RTSRX is set, the RTS output provides the receiver FIFO almost-full condition. When the receiver FIFO backs up, the RTS signal is dropped. The RTS output stalls an external transmitter from delivering data.
D21	R/W	RL	0	**Remote loopback** Provides a remote loopback feature. When set to 1, the TXD transmit output is connected to the RXD receive input. The RL field immediately echoes receive data back as transmit data. Primarily used as test vehicle for external data equipment.
D20	R/W	LL	0	**Local loopback** Provides an internal local loopback feature. When set to 1, the internal receive data stream is connected to the TXD output signal. This field connects the serial channel receiver directly to the serial channel transmitter. Primarily used as a test vehicle for the serial channel driver firmware.
D19:18	R/W	OUT1/OUT2	0	**General-purpose output 1/General-purpose output 2** The OUT1 and OUT2 fields provide extra, miscellaneous serial channel control signal outputs. The OUT1 and OUT2 signals can be routed through PORTA and PORTC pins using PORTA and PORTC special function modes.

Table 85: Serial Channel Control Register A

Bits	Access	Mnemonic	Reset	Description
D17	R/W	DTR	0	**Data terminal ready active** Controls the state of the external data terminal ready signal. ■ Setting DTR to 1 causes the DTR output to go active. ■ Setting DTR to 0 causes the DTR output to go inactive.
D16	R/W	RTS	0	**Request-to-send active** Controls the state of the external request-to-send signal. ■ Setting RTS to 1 causes the RTS output to go active. ■ Setting RTS to 0 causes the RTS output to go inactive.
D15:09	R/W	IE	0	**Receiver interrupt condition** The interrupt enable bits are used to enable an interrupt when the respective status bit is set in Serial Channel Status A. ■ Setting the IE field to 1 enables the interrupt. ■ Setting the IE field to 0 disables the interrupt. Table 86, “Receiver interrupt enable bits,” on page 229, lists individual bit numbers and descriptions.
D08	R/W	ERXDMA	0	**Enable receive DMA requests** Enables the receiver to interact with a DMA channel. When configured to operate in DMA mode, the DMA controller empties the receive data FIFO and delivers the data to memory. The receive status information from Serial Status registers B and C automatically moves to the receive DMA buffer descriptor. Clear this bit to pause the receiver.

Table 85: Serial Channel Control Register A

Bits	Access	Mnemonic	Reset	Description
D07:05	R/W	IE	0	**Receiver interrupt condition** The interrupt enable bits are used to enable an interrupt when the respective status bit is set in Serial Channel Status A. ■ Setting the IE field to 1 enables the interrupt. ■ Setting the IE field to 0 disables the interrupt. Table 86, "Receiver interrupt enable bits," on page 229, lists individual bit numbers and descriptions.
D04:01	R/W	IE	0	**Transmitter interrupt condition** The interrupt enable bits are used to enable an interrupt when the respective status bit is set in Serial Channel Status A. ■ Setting the IE field to 1 enables the interrupt. ■ Setting the IE field to 0 disables the interrupt. Table 87, "Transmitter interrupt enable bits," on page 230, lists individual bit numbers and descriptions.
D00	R/W	ETXDMA	0	**Enable transmit DMA requests** Enables the transmitter to interact with a DMA channel. When configured to operate in DMA mode, the DMA controller loads the transmit data FIFO from memory. Clear this bit to pause the transmitter.

Table 85: Serial Channel Control Register A

Receiver interrupts

Bit	Mnemonic	Description
D15	ERXBRT	Receive break interrupt enable
D14	ERFE	Receive framing error interrupt enable

Table 86: Receiver interrupt enable bits

Bit	Mnemonic	Description
D13	ERXPE	Receive parity error interrupt enable
D12	ERXORUN	Receive overrun interrupt enable
D11	ERXRDY	Receive register ready interrupt enable
D10	ERXHALF	Receive FIFO half-full interrupt enable
D09	ERXBC	Receive buffer closed interrupt enable
D07	ERXDCD	Change in DCD interrupt enable
D06	ERXRI	Change in RI interrupt enable
D05	ERXDSR	Change in DSR interrupt enable

Table 86: Receiver interrupt enable bits

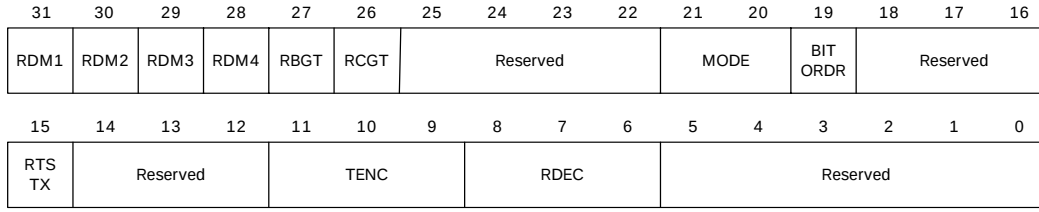
Transmitter interrupts

Bit	Mnemonic	Description
D04	ERXCTS	Change in CTS interrupt enable
D03	ETXRDY	Transmit register empty interrupt enable
D02	ETXHALF	Transmit FIFO half-empty interrupt enable
D01	ERXBC	Transmit buffer closed interrupt enable

Table 87: Transmitter interrupt enable bits

Serial Channel 1, 2 Control Register B

Address: FFD0 0004 / 44



Register bit assignment

Bit	Access	Mnemonic	Reset	Description
D31	R/W	RDM1	0	**Enable receive data match 1/2/3/4**
D30	R/W	RDM2	0	When the serial channel is configured to operate in **UART** mode, the RDM bits enable the receive data match comparators. A receive data match comparison detection can be used to close the current receive buffer descriptor. The last byte in the current receive data buffer contains the match character. Each of these bits enables the respective byte in the Receive Match register.
D29	R/W	RDM3	0	
D28	R/W	RDM4	0	
D27	R/W	RBGT	0	**Enable receive buffer GAP timer** Detects the maximum allowed time from when the first byte is placed into the receive data buffer and when the receive data buffer is closed. When RBGT is set to 1, the BGAP field in Serial Channel Status Register A is set when the timeout value defined in the Receive Buffer GAP Timer register has expired.

Table 88: Serial Channel Control Register B bit definition

Bit	Access	Mnemonic	Reset	Description
D26	R/W	RCGT	0	**Enable receive character GAP timer** Detects the maximum allowed time from when the last byte is placed into the receive data buffer and when the receive data buffer us closed. When RCGT is set to 1, the CGAP field in Serial Channel Status Register A is set when the timeout value defined in the Receive Buffer Character Timer register has expired.
D25:22	N/A	Reserved	N/A	N/A
D21:20	R/W	MODE	0	**SCC mode** 00 UART mode 01 Reserved 10 SPI master mode 11 SPI slave mode Configures the serial channel to operate in UART or SPI modes. The MODE field must be established before the CE bit in Serial Channel Control Register A is set to 1.
D19	R/W	BITORDR	0	**Bit ordering** 0 Normal; transmit/receive LSB (least significant bit) first 1 Reverse; transmit/receive MSB (most significant bit) first Controls the order in which bits are transmitted and received in the Serial Shift register. ■ When BITORDR is set to 0, the bits are processed LSB first, MSB last. ■ When BITORDR is set to 1, the bits are processed MSB first, LSB last.
D18:16	N/A	Reserved	N/A	N/A

Table 88: Serial Channel Control Register B bit definition

Bit	Access	Mnemonic	Reset	Description
D15	R/W	RTSTX	0	**Enable active RTS only while transmitting** **Note:** The RTS bit in Serial Channel Control Register A must be set if setting this bit. Controls the RTS indicator. Do ***not*** set this bit if the RTSRX bit in Serial Channel Control Register A is set. When this bit is set, the RTS output goes active only when the transmitter is actively sending a transmit character. The RTSTX bit allows external hardware to use the RTS signal as a transmit line driver enable signal in multi-drop applications.
D14:12	N/A	Reserved	N/A	N/A
D11:09	R/W	TENC	0	**Transmit encoding**
D08:06	R/W	RDEC	0	**Receive data encoding** The NS7520 can be programmed to encode and decode the serial data stream using one of the eight methods listed here. The transmit and receive coding methods can be selected independently in the transmitter and receiver. ■ **NRZ (000): Default.** A 1 is represented by a high level for the duration of the entire bit-time; a 0 is represented by a low level during the entire bit-time. ■ **NRZB (001).** NRZB is NRZ inverted; that is, a 1 is represented by a low level during the entire bit-time and a 0 is represented by a high level during the bit-time. ■ **NRZI-Mark (010).** A 1 is represented by a transition at the beginning of the bit-time; the level that appears during the previous cell is reversed. A 0 is represented by the absence of a transition at the beginning of the bit cell.

Table 88: Serial Channel Control Register B bit definition

Bit	Access	Mnemonic	Reset	Description
TENC/RDEC *continued*				■ **NRZI-Space (011).** A 0 is represented by a transition at the beginning of the bit-time; the level that appears during the previous cell us reversed. A 1 is represented by the absence of a transition at the beginning of the bit cell. ■ **FM0 (100).** A 0 is represented by a transition at the beginning of the bit-time followed by another transition in the middle of the bit-time. A 0 is represented by a transition only at the beginning of the bit-time. ■ **FM1 (101).** A 1 is represented by a transition at the beginning of the bit-time followed by another transition in the middle of the bit-time. A 0 is represented by a transition only at the beginning of the bit-time. ■ **Manchester (110).** A 1 is represented by a high level during the first half of the bit-time and a low level during the second half of the bit-time. A 0 is represented by a low level during the first half of the bit-time and a high level during the second half of the bit-time. ■ **Differential Manchester (111).** A 1 is represented by a transition at the center of the bit-time, with the opposite polarity from the transition at the center of the preceding bit-time. A 0 is represented by a transition at the center of the bit-time with the same polarity as the transition at the center of the preceding bit-time. In both cases, there are transitions at the beginning of the bit-time to set up the level required to make the correct center transition. See Figure 27, "Data coding example," on page 235 for an illustration of the transmit and receive coding methods.
D05:00	N/A	Reserved	N/A	N/A

Table 88: Serial Channel Control Register B bit definition

Figure 27 shows the different transmit and receive coding methods.

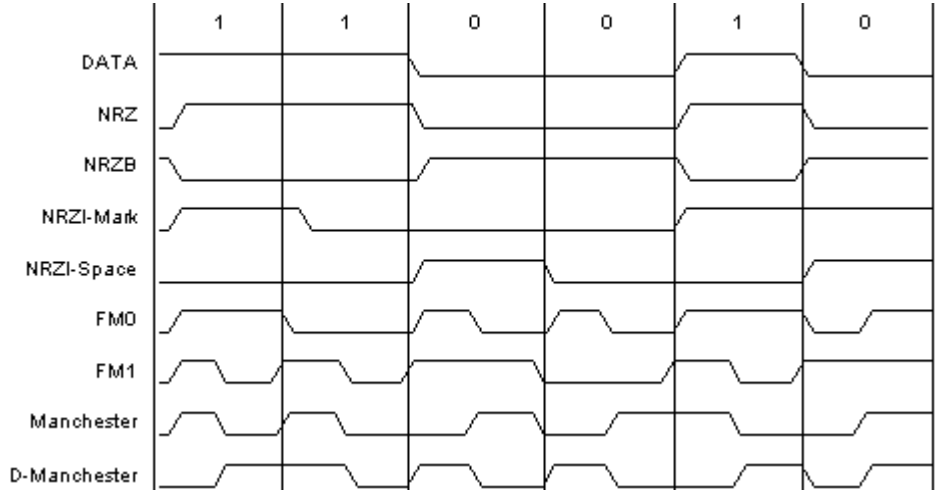


Figure 27: Data coding example

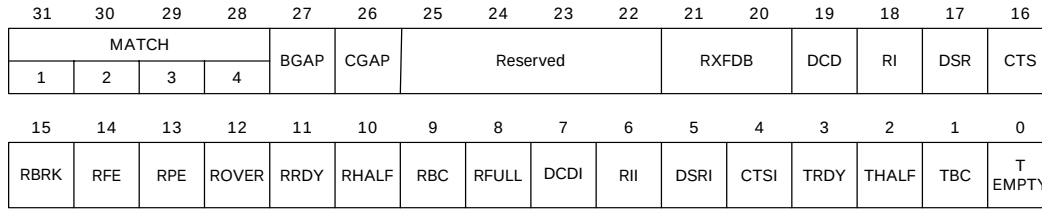
Serial Channel 1, 2 Status Register A

Address: FFD0 0008 / 48

The receive status bits (D31:16) are stuffed into the StatusOrIndex field in the DMA buffer descriptor when DMA operations are enabled and a buffer is closed. The upper six bits of this register (D31:26) can be cleared by writing a 1 to the respective bit position (used when interrupt-driven). These six bits are updated automatically when a receive buffer is closed.

The receive interrupt pending bits (D15:00) are cleared by writing a 1 to the respective bit position.

Note: All status bits are active high unless an underscore (_) appears in the signal name; in this case, they are active low.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R	MATCH1	0	**Character Match1**
D30	R	MATCH2	0	**Character Match2**
D29	R	MATCH3	0	**Character Match3**
D28	R	MATCH4	0	**Character Match4** Set when a MATCH character is configured in the Receive Match register at the same time the enable receive data match bit is set in Serial Channel Control Register B. The MATCH bit indicates that a data match was found in the receive data stream, and the current receive data buffer has been closed. The last character in the receive data buffer contains the actual MATCH character. When the receiver is configured for DMA operation, the MATCH status bits are written automatically to the DMA receive buffer descriptor's status field. When not using DMA, the MATCH fields are valid only while the RBC bit in this register is set.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D27	R	BGAP	0	**Buffer GAP timer** Set when the enable receive buffer gap timer is set in Serial Channel Control Register B and the timeout value defined in the Receive Buffer GAP Timer register has expired. This bit indicates that the maximum allowed time has occurred since the first byte was placed into the receive data buffer. The receive data buffer is closed under this condition. When the receiver is configured for DMA, the BGAP bit is written automatically to the DMA receive buffer descriptor's status field. When not using DMA, BGAP is valid only while the RBC bit in this register is set.
D26	R	CGAP	0	**Character GAP timer** Set when the enable receive character gap timer is set in Serial Channel Control Register B and the timeout value defined in the Receive Character GAP Timer register has expired. This bit indicates that the maximum allowed time has occurred since the previous byte was placed into the receive data buffer. The receive data buffer is closed under this condition. When the receiver is configured for DMA, the CGAP bit is written automatically to the DMA buffer descriptor's status field. When not using DMA, CGAP is valid only while the RBC bit is set in this register.
D25:22	N/A	Reserved	N/A	N/A

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D21:20	R	RXFDB	0	**Receive FIFO data available** 00 Full-word 01 One byte 10 Half-word 11 Three bytes (LENDIAN determines which three) Identifies the number of valid bytes contained in the next long word to be read from the Serial Channel FIFO Data register. The next read of the FIFO can contain one, two, three, or four valid bytes of data. This field must be read before the FIFO is read, to determine which bytes of the 4-byte long word contain valid data. Normal Endian byte-ordering rules apply to the Serial Channel FIFO Data register.
D19	R	DCD	0	**Current data carrier detect state** 0 Inactive 1 Active Identifies the current state of the EIA data carrier detect signal.
D18	R	RI	0	**Current ring indicator state** 0 Inactive 1 Active Indicates the current state of the EIA ring indicator signal.
D17	R	DSR	0	**Current data set ready state** 0 Inactive 1 Active Indicate the current state if the EIA data set ready signal.
D16	R	CTS	0	**Current clear to send state** 0 Inactive 1 Active Identifies the current state of the EIA clear to send signal.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D15	R/C	RBRK	0	**Receive break interrupt pending** Indicates that a UART receive break condition has been found. Once set, the RBRK bit remains set until acknowledged. RBRK is acknowledged by writing a 1 to this same bit position in this register. The RBRK status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D14	R/C	RFE	0	**Receive framing error interrupt pending** Indicates that a receive framing error condition has been found. Once set, the RFE bit remains set until acknowledged. RFE is acknowledged by writing a 1 to this same bit position in this register. The RFE status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D13	R/C	RPE	0	**Receive parity error interrupt pending** Indicates that a receive parity error condition has been found. Once set, the RPE field remains set until acknowledged. RPE is acknowledged by writing to this same bit position in this register. The RPE status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D12	R/C	ROVER	0	**Receive overrun interrupt pending** Indicates that a receive overrun error condition has been found. An overrun condition indicates that the FIFO was full while data needed to be written by the receiver. When the FIFO is full, any new receive data will be discarded; the contents of the FIFO before the overrun condition remains the same. Once set, the ROVER field remains set until acknowledged. ROVER is acknowledged by writing a 1 to this same bit position in this register. The ROVER status condition can be programmed to generate an interrupt by setting the related IE bit in the Serial Channel Control Register A.
D11	R	RRDY	0	**Receive register ready interrupt pending** Indicates that data is available to be read from the FIFO Data register. Before reading the FIFO Data register, the RXFDB field in this register must be read to determine how many active bytes are available during the next read of the FIFO Data register. RRDY typically is used only in interrupt-driven applications; it is not used for DMA operation. The RRDY status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A. RRDY is never active while RBC is active. The RBC bit must be acknowledged to activate RRDY. When the receiver is configured to operate in DMA mode, the interlock between RBC and RRDY is handled automatically in hardware.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D10	R	RHALF	0	**Receive FIFO half-full interrupt pending** Indicates that the receive data FIFO contains at least 16 bytes. RHALF typically is used only in interrupt-driven applications; it is not used for DMA operation. The RHALF status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D09	R/C	RBC	0	**Receive buffer closed interrupt pending** Indicates a receive buffer closed condition. Once set, the RBC bit remains set until acknowledged. RBC is acknowledged by writing a 1 to this same bit position in this register. The RBC bit is acknowledged automatically by hardware when the receiver is configured to operate in DMA mode. The RBC status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A. The RBC field indicates that bits D31:26 in this register are valid. While the RBC field is active, the RRDY field is not. To activate RRDY (to read the data FIFO), the RBC bit must be acknowledged. This interlock between RBC and RRDY allows the firmware driver to read the D31:26 status bits. When operating in DMA mode, the status bits are written automatically to the receive DMA buffer descriptor, and the interlock between RBC and RRDY is handled automatically in the hardware.
D08	R	RFULL	0	**Receive FIFO full** Indicates that the receive data FIFO currently is full.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D07	R/C	DCDI	0	**Change in DCD interrupt pending** Indicates a state change in the EIA data carrier detect signal. Once set, the DCDI field remains set until acknowledged. DCDI is acknowledged by writing a 1 to this same bit position in this register. The DCDI status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D06	R/C	RII	0	**Change in RI interrupt pending** Indicates a state change in the EIA ring indicator signal. Once set, the RII bit remains set until acknowledged. RII is acknowledged by writing a 1 to this same bit position in this register. The RII status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D05	R/C	DSRI	0	**Change in DSR interrupt pending** Indicates a state change in the EIA data set ready signal. Once set, the DSRI bit remains set until acknowledged. DSRI is acknowledged by writing a 1 to this same bit position in this register. The DSRI state condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.
D04	R/C	CTSI	0	**Change in CTS interrupt pending** Indicates a state change in the EIA clear to send signal. Once set, the CTSI bit remains set until acknowledged. CTSI is acknowledged by writing a 1 to this same bit position in this register. The CTSI state condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
D03	R	TRDY	0	**Transmit register empty interrupt pending** Indicates data can be written to the FIFO Data register. TRDY typically is used only in interrupt-driven applications; it is not used for DMA operation. The TRDY status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A. TRDY is never active while the TBC bit is active. TBC must be acknowledged to activate the TRDY bit. When the transmitter is configured to operate in DMA mode, the interlock between TBC and TRDY is handled automatically in hardware.
D02	R	THALF	0	**Transmit FIFO half-empty interrupt pending** Indicates that the transmit data FIFO contains room for at least 16 bytes. The THALF field typically is used only in interrupt-driven applications; it is not used for DMA operation. The THALF status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel COntrol Register A.
D01	R/C	TBC	0	**Transmit buffer closed interrupt pending** Indicates a transmit buffer closed condition. Once set, the TBC bit remains set until acknowledged. TBC is acknowledged by writing a 1 to this same bit position in this register. The TBC bit is acknowledged automatically by hardware when the transmitter is configured to operate in DMA mode. The TBC status condition can be programmed to generate an interrupt by setting the related IE bit in Serial Channel Control Register A.

Table 89: Serial Channel Status Register A bit definition

Bits	Access	Mnemonic	Reset	Description
TBC *continued*				The TBC field is set when the last character in the transmit FIFO has been shifted out of the shift register and the FIFO currently is empty. While the TBC field is active, the TRDY bit is not active. To activate TRDY (to write to the data FIFO), TBC must be acknowledged. When operating in DMA mode, the interlock between TBC and TRDY is handled automatically in hardware.
D00	R	TEMPTY	0	**Transmit FIFO empty** Indicates that the transmit data FIFO is currently empty. TEMPTY simply reports the status of the FIFO; it does not indicate that the character currently in the transmit shift register has been transmitted. The TBC bit must be used for the *all sent* condition.

Table 89: Serial Channel Status Register A bit definition

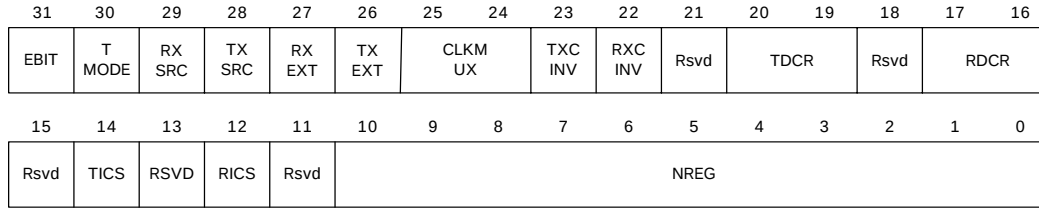
Serial Channel 1, 2 Bit-Rate registers

Address: FFD0 000C / 4C

The serial channel bit rate registers configure the bit-rate for each serial channel.

The serial channel can be configured to operate using an internal or external timing reference. When configured for internal timing, the timing reference is provided by the bit-rate generator. When configured for external timing, the timing reference is provided by the PORTC signals RXCLK and TXCLK.

The serial channel can be configured to operate in either 1X (synchronous) mode or 16X (asynchronous) mode. When using the internal bit-rate generator, the frequency must be configured properly to match the mode being used.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	EBIT	0	**Bit-rate generator enable** Enables or disables the internal bit-rate generator. 0 Disables the bit-rate generator 1 Enables the bit-rate generator
D30	R/W	TMODE	0	**Timing mode** 0 16X mode 1 Use TDCR/RDCR ■ When TMODE is set to 0, the serial channel is forced to 16X operation. ■ When TMODE is set to 1, an oversampling multiplier is chosen by the TDCR and RDCR fields. When the transmitter or receiver are configured to use an external timing source (TXSRC or RXSRC set to 1), the serial channel is configured automatically to operate in 1X mode. The serial channel clock is provided directly from the OUT1/ OUT2 inputs through the PORTA/PORTC interface. Setting TXSRC to 1 overrides any TXEXT setting.
D29	R/W	RXSRC	0	**Receive clock source** 0 Internal 1 External (input through OUT1 signal) Controls the source of the receiver clock. The receive clock can be provided by an internal source, as determined by the value in the RICS field, or by an input on the OUT1 signal attached to the PORTA/C ports (configured as a special function input).

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D28	R/W	TXSRC	0	**Transmit clock source** 0 Internal 1 External (input through OUT2 signal) Controls the source of the transmitter clock. The transmitter clock can be provided by an internal source, as determined by the value in the TICS field, or by an input on the OUT2 signal attached to the PORTC pin (configured as a special function input).
D27	R/W	RXEXT	0	**Drive receive clock external** 0 Disable 1 Enable; drive RXCLK out OUT1 signal at PORTA/PORTC Enables the receiver clock to be driven on the OUT1 signal attached to the PORTA/PORTC ports. When using the OUT1 signal, the PORTA/PORTC pin must be configured as special function output.
D26	R/W	TXEXT	0	**Drive transmit clock external** 0 Disable 1 Enable; drive TXCLK out OUT2 signal at PORTC Enables the transmitter clock to be driven on the OUT2 signal attached to PORTC port. When using the OUT2 signal, The PORTC pin must be configured as special function output.

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D25:24	R/W	CLKMUX	0	**BRG input clock** 00 Input clock defined by F_{XTALE} 01 Input clock defined by F_{SYSCLK} 10 Input clock defined by input on OUT1 11 Input clock defined by input on OUT2 Controls the bit-rate generator clock source. The bit-rate generator can use one of four clock source: the external oscillator, the internal PLL SYSCLK output, an input signal on the OUT1 signal on PORTA/PORTC, or an input signal on the OUT2 signal attached to PORTC. When using either OUT1 or OUT2, the PORTA/PORTC port pin must be configured as special function input.
D23	R/W	TXCINV	0	**Transmit clock invert** 0 Normal; TXD driven on falling edge of TX clock 1 Inverted; TXD driven on rising edge of TX clock Controls the relationship between transmit clock and transmit data. ■ When set to 0, transmit data changes relative to the high-to-low transition of the transmit clock. ■ When set to 1, transmit data changes relative to the low-to-high transition of the transmit clock. **Note:** When using SPI mode, this bit must be set to zero.

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D22	R/W	RXCINV	0	**Receive clock invert** 0 Normal; RXD sampled on rising edge of RX clock 1 Inverted; RXD sampled on falling edge of RX clock Controls the relationship between receive clock and receive data. ■ When set to 0, receive data input is sampled at the low-to-high transition of the receive clock. ■ When set to 1, receive data input is sampled at the high-to-low transition of the receive clock. **Note:** When using SPI mode, this bit must be set to zero.
D21	N/A	Reserved	N/A	N/A

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D20:19	R/W	TDCR	0	**Transmit divide clock rate** 00 1x clock mode (only NRZ or NRZI allowed) 01 8x clock mode 10 16x clock mode 11 32x clock mode Determine the oversampling multiplier for the transmitter and receiver clocks. ■ **If DPLL is not used and you are not using UART,** select the value 1x. ■ **If DPLL is not used but you are using UART,** select 8x, 16x, or 32x (16x is recommended). In most applications, the TDCR and RDCR configurations should match. ■ **When DPLL is used in the application,** the selected TDCR/RDCR value is a function of the transmitter encoding. NRZ and NRZI modes can use the 1x configuration; all other encoding must use 8x, 16x, or 32x configuration mode. 8x configuration provides the highest data rate; 32x mode provides the highest resolution. The TMODE bit in the Serial Channel Bit-Rate register is maintained for NET + ARM family backward compatibility. When setting TDCR or RDCR to a non-zero value, the TMODE must be set to 1. When the TMODE, TDCR, and RDCR fields are set to 0, the port defaults to 16x mode of operation.
D18	N/A	Reserved	N/A	N/A

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D17:16	R/W	RDCR	0	**Receive divide clock rate** 00 1x clock mode (only NRZ or NRZI allowed) 01 8x clock mode 10 16x clock mode 11 32x clock mode Determine the oversampling multiplier for the transmitter and receiver clocks. ■ **If DPLL is not used and you are not using UART,** select the value 1x. ■ **If DPLL is not used but you are using UART,** select 8x, 16x, or 32x (16x is recommended). In most applications, the TDCR and RDCR configurations should match. ■ **When DPLL is used in the application,** the selected TDCR/RDCR value is a function of the transmitter encoding. NRZ and NRZI modes can use the 1x configuration; all other encoding must use 8x, 16x, or 32x configuration mode. 8x configuration provides the highest data rate; 32x mode provides the highest resolution. The TMODE bit in the Serial Channel Bit-Rate register is maintained for NET + ARM family backward compatibility. When setting TDCR or RDCR to a non-zero value, the TMODE must be set to 1. When the TMODE, TDCR, and RDCR fields are set to 0, the port defaults to 16x mode of operation.
D15	N/A	Reserved	N/A	N/A

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D14	R/W	TICS	0	**Transmit internal clock source** 0 BRG; the transmitter uses BRG output for its clock 1 DPLL; the transmitter uses the extracted clock provided by the DPLL. When the TXSRC field is set to 0, the transmitter operates using an internal clock. There are two sources for internal clocks: the bit-rate generator (BRG) and the receiver digital phase lock loop (DPLL). The BRG uses a divider mechanism for clock generation. The DPLL extracts the clock from the incoming receive data stream.
D13	N/A	Reserved	N/A	N/A
D12	R/W	RICS	0	**Receiver internal clock source** 0 BRG; the transmitter uses BRG output for its clock 1 DPLL; the receiver uses the extracted clock provided by the DPLL. When the RXSRC field is set to 0, the receiver operates using an internal clock. There are two sources for internal clocks: the bit-rate generator (BRG) and the receiver digital phase lock loop (DPLL). The BRG uses a divider mechanism for clock generation. The DPLL extracts the clock from the incoming receive data stream.
D11	N/A	Reserved	N/A	N/A

Table 90: Serial Channel Bit-Rate register bit definition

Bits	Access	Mnemonic	Reset	Description
D10:00	R/W	NREG	0	**N register** The N register value is determined by the following equations. F_{BRG} must be adjusted according to the TMODE and TDCR/RDCR settings. For a 16x setting, for example, F_{BRG} is 16 times the desired baud rate. $F_{BRG} = F_{XTALE}/[2*(N+1)$ Using a CLKMUX setting of 00 $F_{BRG} = F_{SYSCLK}/[2*(N+1)$ Using a CLKMUX setting of 01 $F_{BRG} = F_{OUT1}/[2*(N+1)$ Using a CLKMUX setting of 10 $F_{BRG} = F_{OUT2}/[2*(N+1)$ Using a CLKMUX setting of 11 The maximum value for F_{OUT1} and F_{OUT2} is $F_{SYSCLK}/4$. See Table 91: "Bit rate examples" on page 254 for sample bit rates.

Table 90: Serial Channel Bit-Rate register bit definition

Max baudrates with different clock sources

Max baud rates for the serial port depend on the clock source you are using.

With the 18.432MHz crystal using XTALE as the clock source:
FXTALE = xtal/5 = 18.432MHz/5 = 3.6864MHz

This does not change with speed grade.

- **16X**

```
Fbrg = FXTALE/[2 * 16 * (N+1)]
115200 = 3686400/2*16*(N+1)          or
2*16*(N+1) = 3686400/115200          or
2*16*(N+1) = 32                      or
N = 32/32-1 = 0
Max baudrate @ 16X = 115.2K
```

- **8X**

```
Fbrg = FXTALE/[2 * 8 * (N+1)]
230400 = 3686400/2*8*(N+1)          or
2*8*(N+1) = 3686400/230400          or
2*8*(N+1) = 16                      or
N = 16/16-1 = 0
Max baudrate @ 8X = 230.4K
```

With the 18.432MHz crystal using SYSCLK as the clock source:
Max. baudrate depends on speed grade. Max. *programmable* baudrate = SYSCLK/32; max *recommended* baudrate is 922K.

- **16X @ 55.296MHz**

```
Fbrg = SYSCLK/[2 * 16 * (N+1)]
1728000 = 55296000/2*16*(N+1)       or
2*16*(N+1) = 55296000/1728000       or
2*16*(N+1) = 32                     or
N = 32/32-1 = 0
Max programmable baudrate = 1,728K
```

- **16X @ 46.08MHz**

```
Fbrg = SYSCLK/[2 * 16 * (N+1)]
1440000 = 46080000/2*16*(N+1)       or
2*16*(N+1) = 46080000/1440000       or
2*16*(N+1) = 32                     or
N = 32/32-1 = 0
Max programmable baudrate = 1,440K
```

- **16X @ 36.864MHz**

```
Fbrg = SYSCLK/[2 * 16 * (N+1)]
1152000 = 36864000/2*16*(N+1)       or
2*16*(N+1) = 36864000/1152000       or
2*16*(N+1) = 16                     or
N = 32/32-1 = 0
Max programmable baudrate = 1,152K
```

These bit rate examples are generated by a bit-rate generator using an 18.432 MHz quartz crystal.

		N register		
Bit rate	**CLKMUX**	**X1 mode**	**X8 mode**	**X16 mode**
75	00	N/A	N/A	1535
150	00	N/A	N/A	767
300	00	N/A	N/A	383
600	00	N/A	N/A	191
1200	00	1535	N/A	95
2400	00	767	N/A	47
4800	00	383	N/A	23
7200	00	255	N/A	15
9600	00	191	N/A	11
14400	00	127	N/A	7
19200	00	95	N/A	5
28800	00	63	N/A	3
38400	00	47	N/A	2
57600	00	31	N/A	1
115200	00	15	N/A	0
230400	00	N/A	0	N/A

Table 91: Bit rate examples

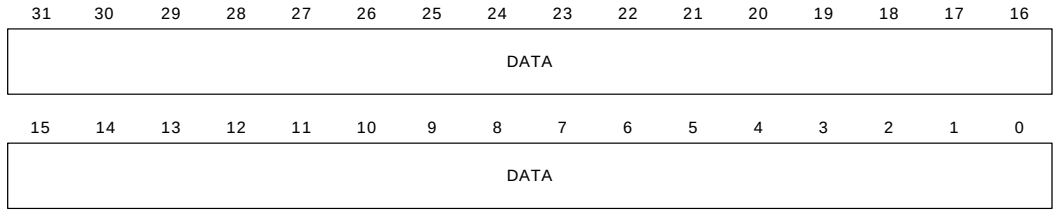
Serial Channel 1, 2 FIFO registers

Address: FFD0 0010 / 50

The serial channel FIFO data registers manually interface with the serial controller FIFOs instead of using DMA support.

Writing to this register loads the transmit FIFO. This register can be written only when the TRDY bit is set in Serial Channel Status Register A. Writing to the Serial Channel FIFO register automatically clears the TRDY bit.

Reading to this register empties the receive FIFO. Data is available when the RRDY bit is set in Serial Channel Status Register A. The RXFDB bits in Serial Channel Status Register A indicate how many bytes are available to be read. Reading the Serial Channel FIFO register automatically clears the RRDY bit.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	DATA	0	**Data** Writing to this register loads the transmit FIFO Reading to this register empties the receive FIFO.

Serial Channel 1, 2 Receive Buffer Gap Timer

Address: FFD0 0014 / 54

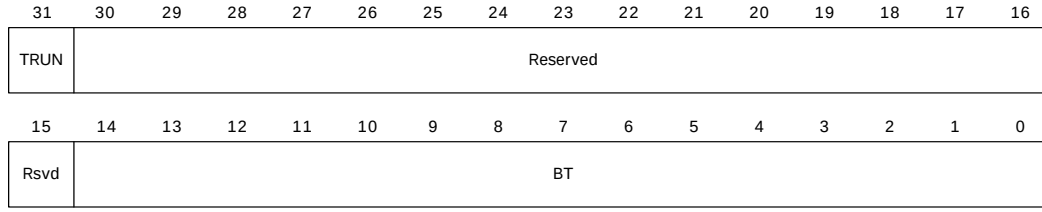
The Receive Buffer Gap Timer register closes out a receive serial data buffer. The timer is reset when the first character is received in a new buffer. New characters are received while the timer operates. When the timer reaches its programmed threshold, the receive data buffer is closed.

If the serial channel is configured to operate in DMA mode, the DMA channel is signaled to close the buffer and start a new buffer. If the serial channel is configured to operate in interrupt mode, the expiration of the timer causes an interrupt to be generated.

The receive buffer timer uses F_{XTALE} and a 9-bit prescaler within the SER module. The receive buffer timer is configured with a 15-bit programmable counter. The effective buffer timer value is defined by this equation:

$TIMEOUT = [512 * (BT + 1)] / F_{XTALE}$

Register diagram and bit assignment



Bits	Access	Mnemonic	Reset	Description
D31	R/W	TRUN	0	**Enable timer to run** Set to 1 to allow the receive buffer gap timer to operate.
D30:15	N/A	Reserved	N/A	N/A
D14:00	R/W	BT	0	**BT timer** The required value for the receive buffer gap timer is a function of the channel bit-rate and the maximum receive buffer size. It is recommended that you set the buffer gap timer to a value that is slightly larger that the amount of time required to fill the maximum buffer size using the channel bit-rate. Use this equation to define the recommended buffer gap timer value: `BT RECOMMENDED = [(MAX_RX_BUF_SIZE * 1.10 * F`$_{XTALE}$`) / (bit-rate * 512)] - 1` If the resulting value doesn't fit within the range for BT, reconsider the size of `MAX_RF_BUF_SIZE`.

Table 92: Serial Channel Receive Buffer Gap Timer bit definition

Serial Channel 1, 2 Receive Character Gap Timer

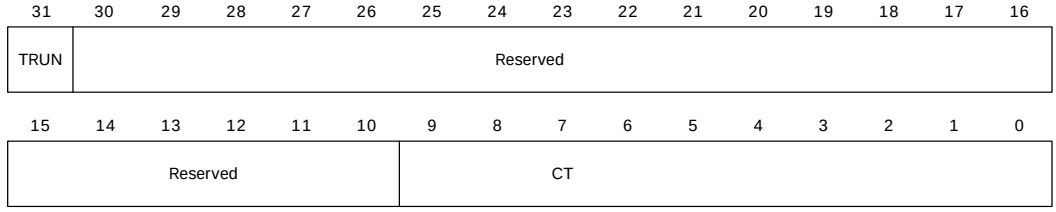
Address: FFD0 0018 / 58

The receive character gap timer closes out a receive serial data buffer due to a gap between characters. The timer is reset when a character is received. When the timer reaches its programmed threshold, the receive data buffer is closed.

If the serial channel is configured to operate in DMA mode, the DMA channel is signaled to close the buffer and start a new buffer. If the serial channel is configured in interrupt mode, the expiration of the timer causes an interrupt to be generated.

The receive character timer uses F_{XTALE} and a 9-bit prescaler within the SER module. The receive character timer is configured with a 10-bit programmable counter. The effective buffer timer value is defined by this equation:

$TIMEOUT = [512 * (CT + 10)] / F_{XTALE}$



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	TRUN	0	**Enable timer to run** Set to 1 to allow the receive character gap timer to operate.
D30:10	N/A	Reserved	N/A	N/A
D09:00	R/W	CT	0	**CT value** The required value for the receive character timer is a function of the channel bit-rate. It is recommended that you set the character timer to a value that is 10 times the character period for the channel bit-rate. Use this equation to define the recommended character timer value: $CT\ RECOMMENDED = [(10 * F_{XTALE}) / (bit\text{-}rate * 512)] - 1$

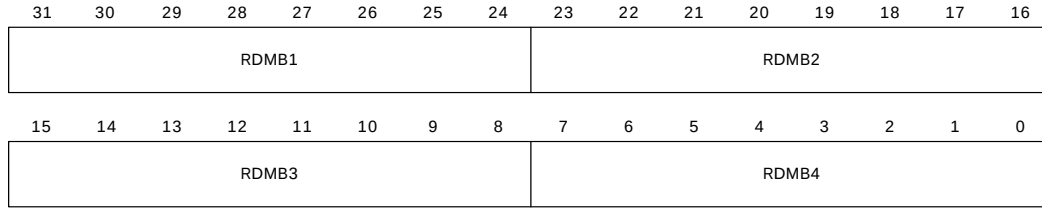
Table 93: Serial Channel Receive Character Gap Timer bit definition

Serial Channel 1,2 Receive Match register

Address: FFD0 001C / 5C

When the serial channel is configured for UART mode, the Receive Match register provides the data bytes that the receiver uses to compare against the incoming receive data stream. If a match is found and the appropriate match enable bit is set in Serial Channel Control Register B, the current receive data buffer is closed by the serial channel and a new buffer is started.

In UART configurations, individual bits within the match register bytes can be masked using the Receive Match MASK register (see "Serial Channel 1, 2 Receive Match MASK register" on page 258).



Register bit assignment

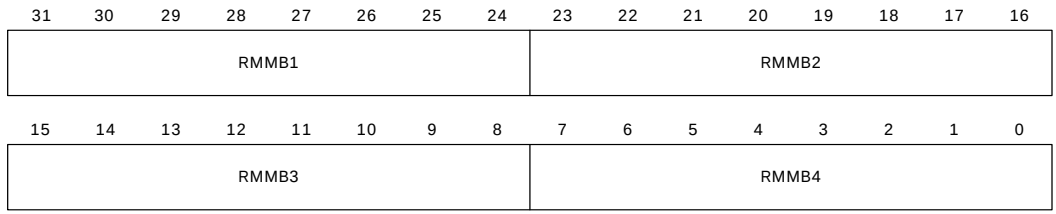
Bits	Access	Mnemonic	Reset	Description
D31:24	R/W	RDMB1	0	Receive data match byte 1
D23:16	R/W	RDMB2	0	Receive data match byte 2
D15:08	R/W	RDMB3	0	Receive data match byte 3
D07:00	R/W	RDMB4	0	Receive data match byte 4

Table 94: Serial Channel Receive Match register bit definition

Serial Channel 1, 2 Receive Match MASK register

Address: FFD0 0020 / 60

The Receive Match MASK register masks those bits in the Receive Match Data register that should *not* be included in the match comparison. To mask a bit in the match comparison function, place a 1 in the same bit position in this register.



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:24	R/W	RMMB1	0	Receive mask match byte 1
D23:16	R/W	RMMB2	0	Receive mask match byte 2
D15:08	R/W	RMMB3	0	Receive mask match byte 3
D07:00	R/W	RMMB4	0	Receive mask match byte 4

Table 95: Serial Channel Receive Match MASK register bit definition

Electrical Characteristics

CHAPTER 11

This chapter provides the electrical specifications and timing relationships integral to NS7520 operation. Electrical specifications include DC and AC characteristics.

DC characteristics

DC electrical specifications define the power supply voltages and currents, and the I/O voltage and drive characteristics.

Recommended operating conditions

The NS7520 operates using an internal core V_{DD} supply voltage of 1.5V. A 3.3V supply is required for the I/O cells, which drive/accept 3.3V levels.

This table defines the DC operating (thermal) conditions for the NS7520. Operating the NS7520 outside these conditions results in unpredictable behavior.

Sym	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Core supply voltage		1.4	1.5	1.6	V
V_{CC}	I/O supply voltage		3.0	3.3	3.6	V
T_{OP}	Ambient temperature		-40		85	^{o}C
T_J	Junction temperature			110		^{o}C
T_{STG}	Storage temperature		-40		125	^{o}C
θ_J	Pkg thermal resistance			50		^{o}C/W
I_{IH}	Input threshold	No pullup	-10		10	µA
I_{IL}	Input current as “0”	No pullup	10		10	µA
I_{OZ}	HighZ leakage current	Any input	-10		10	µA
C_{IO}	Pin capacitance	$V_O = 0$			7	pF

Table 96: Recommended operating (thermal) conditions

Input/Output characteristics

Table 97 shows DC characteristics for inputs. Table 98 shows DC characteristics for outputs.

Sym	Parameter	Min	Max	Unit
V_{IH}	Input high voltage	2.0	3.6	V
V_{IL}	Input low voltage	V_{SS} – 0.3	0.8	V

Table 97: DC characteristics — inputs

Sym	Parameter	Condition		Min	Max	Unit
P	Power consumption	F_{SYSCLK} = 55 MHz			508	mW
			Core		192	mW
			I/O		316	mW
		F_{SYSCLK} = 46 MHz			425	mW
			Core		161	mW
			I/O		264	mW
		F_{SYSCLK} = 36 MHz			333	mW
			Core		126	mW
			I/O		207	mW
V_{OL}	Output low voltage	Outputs and bi-directional		0	0.4	V
V_{OH}	Output high voltage	Outputs and bi-directional		2.4	V_{DD}	V

Table 98: DC characteristics — outputs

Pad pullup and pulldown characteristics

Figure 28 illustrates the characteristics for a pad with internal pullup; Figure 29 illustrates the characteristics for a pad with internal pulldown. See "Pinout detail tables and signal descriptions," beginning on page 11, for information about which pins use pullup and pulldown resistors.

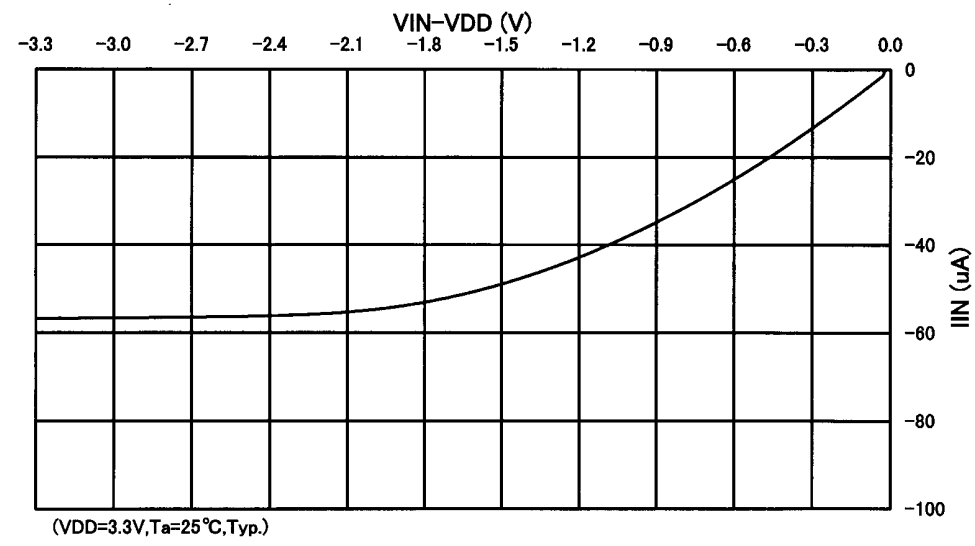


Figure 28: Internal pullup characteristics

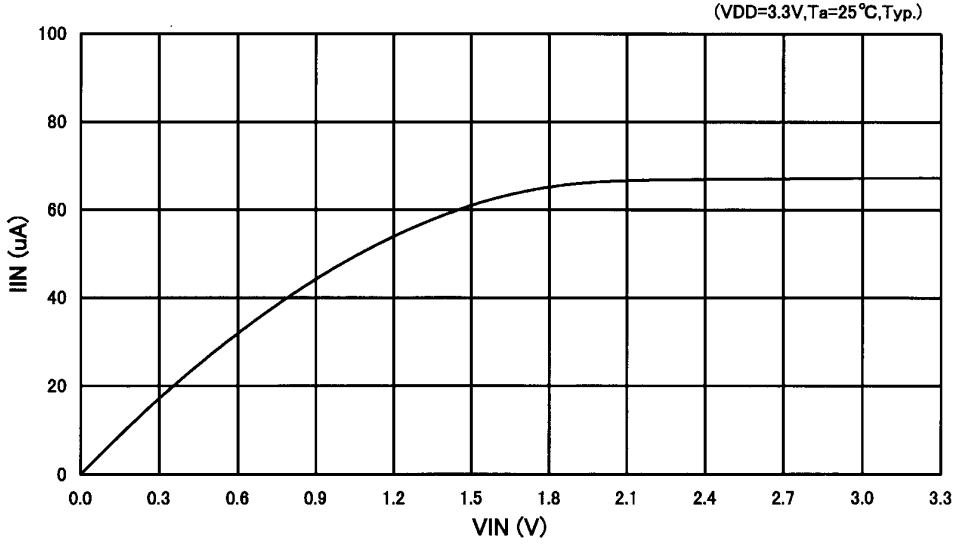


Figure 29: Internal pulldown characteristics

Absolute maximum ratings

Table 99 defines the maximum values for the voltages that the NS7520 can withstand without being damaged.

Sym	Parameter	Min	Max
V_{DD}	Core supply voltage	-0.3	3.15
V_{CC}	I/O supply voltage	-0.3	3.9
V_{IN}	Input voltage	-0.3	3.9
V_{OUT}	Output voltage	-0.3	3.9

Table 99: Maximum voltage ratings

AC characteristics

AC electrical specifications define the timing relationship between signals for interfaces and modes within a given interface.

AC electrical specifications

The AC electrical specifications are based on the system configuration shown in Figure 30, with a 5pF allowance for PCB capacitance and a 0.25 ns allowance for PCB delay. The timing of the buffers, SDRAM, and the like must be added to complete timing analysis. In systems where SDRAM is not used, two devices are expected to replace the SDRAMs shown in Figure 30; that is, they are tied directly to the chip. System loading information is shown in Table 100: "System loading details" on page 266.

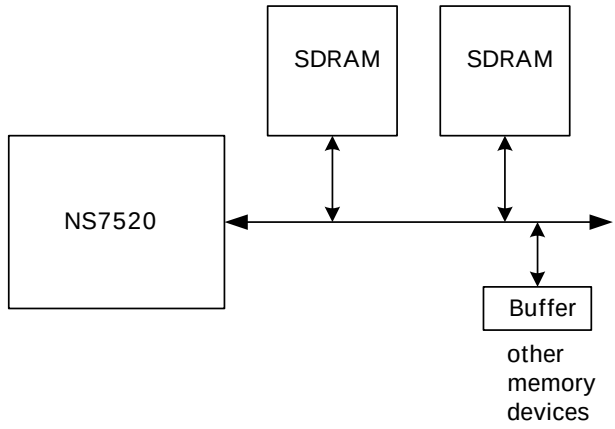


Figure 30: System configuration for specified timing

Signal	Estimated load (pF)	Device loads
BCLK	23	Two SDRAMs, 1 clock buffer/ clock input to PLD
A[27:0], CAS[3:0]_	23	Two SDRAM A*n*, 1 buffer/PLD
CS[4:0]_	13	Two SDRAM CS*n*, 1 buffer PLD
DATA[31:0]	18	One SDRAM DQ, 1 buffer/PLD
BE*_	19	One SDRAM DQ, 1 buffer/PLD
TS_, TA_, TEA_, BR_, BG_, BUSY_, WE_, OE_	15	1 buffer/PLD
PORTA3, PORTA1, PORTC3, PORTC1 (operating external DMA)	15	1 buffer/PLD
Other PORTA[*] and PORTC[*], TDO	85	Tester load
MDC, MDIO, TXEN, TXER, TXD[3:0]	20	One PHY

Table 100: System loading details

Exceeding the loading shown in Table 100 can result in additional signal delay. The delay can be approximated by derating the output buffer based on the expected load capacitance per the values shown in Table 101.

Signal	Derating (ns/pF)
BCLK	0.069
A[27:0], TS_, TA_, TEA_, BR_, BG_, BUSY_, DATA[31:0]	0.150
BE[3:0]	0.300
CS[4:0]_, CAS[3:0], RW_, WE_, OE_	0.137
MDC, TXD[3:0], TXER, TXEN, TDO	0.274

Table 101: Output buffer derating by load capacitance

Oscillator Characteristics

Figure 31 illustrates the recommended oscillator circuit details.

- **Rise/fall time.** The max rise/fall time on the system clock input pin is 1.5ns when used with an external oscillator.
- **Duty cycle.** The duty cycle is system-dependent with an external oscillator. It affects the setup and hold times of signals that change in the falling clock edges, such as WE_/OE_.

  Recommendation: Use a 3.3V, 50±10% duty cycle oscillator with a 100 ohm series resistor at the output. The PLLs can handle a 25% duty cycle clock (minimum high/low time 4.5nS).

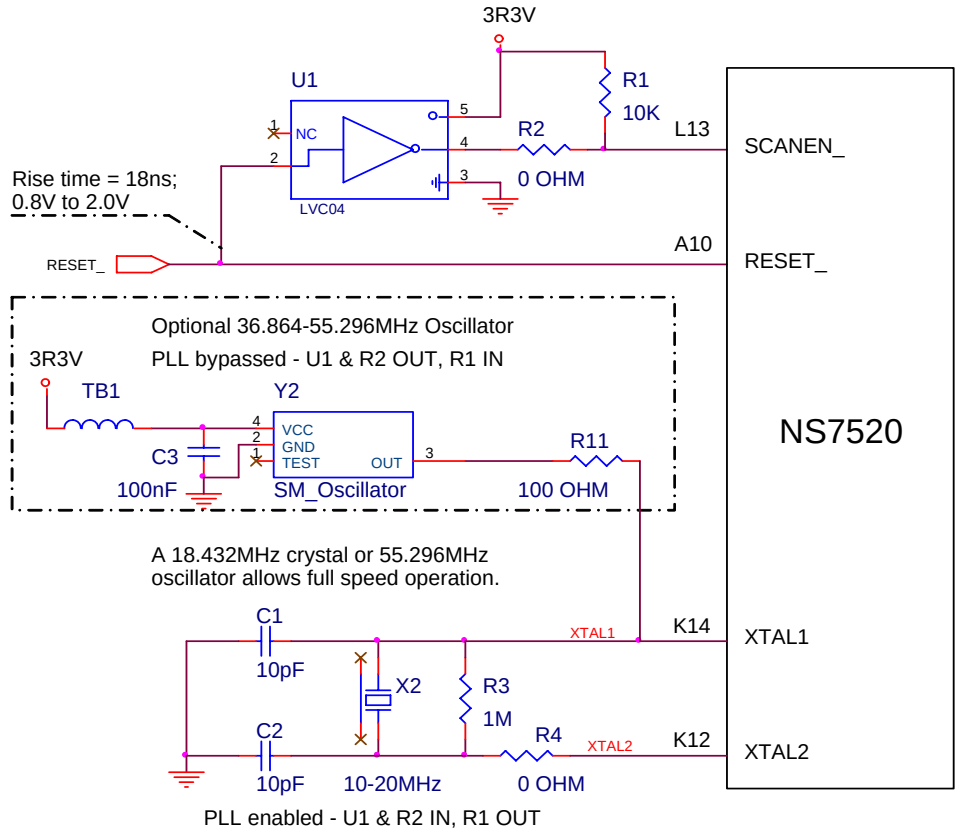


Figure 31: Oscillator circuit details

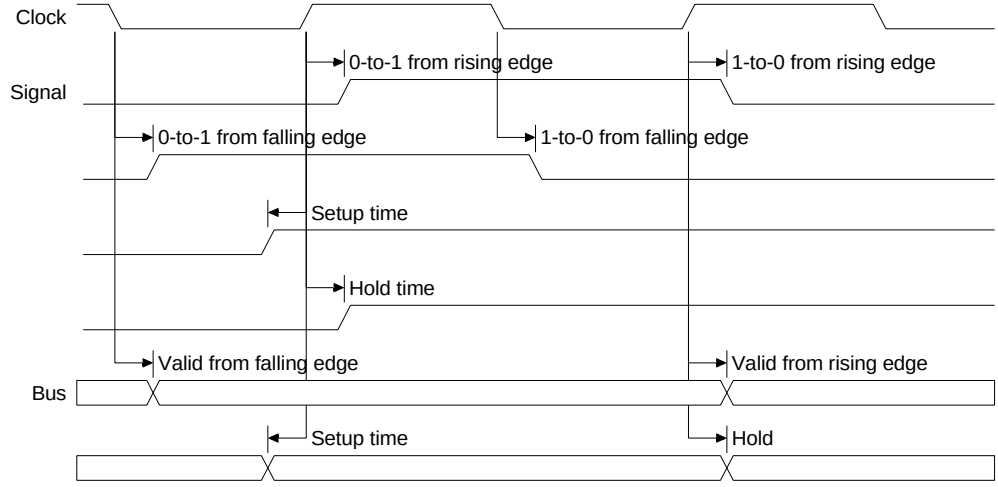
Timing Diagrams

Timing_Specifications

All timing specifications consist of the relationship between a reference clock and a signal:

- There are bussed and non–bussed signals. Non–bussed signals separately illustrate 0–to–1 and 1–to–0 transitions.
- Inputs have setup/hold times versus clock rising.
- Outputs have switching time relative to either clock rising or clock falling.

Note: Timing relationships in this diagram are drawn without proportion to actual delay.



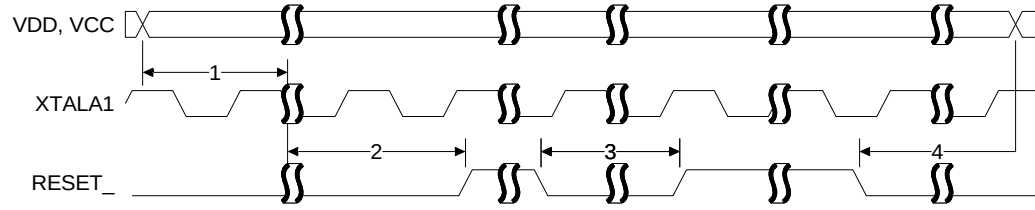
Reset_timing

From a cold start, RESET_ must be asserted until all power supplies are above their specified thresholds. An additional 8 microseconds is required for oscillator settling time (allow 40ms for crystal startup).

Due to an internal three flip-flop delay on the external RESET_ signal, after the oscillator is settled, RESET_ must be asserted for three periods of the XTALA1 clock in these situations:

- Before release of reset after application of power
- While valid power is maintained to initiate *hot reset* (reset while power is at or above specified thresholds)
- Before loss of valid power during power outage/power down

The PORTC4 output indicates the reset state of the chip. PORTC4 persists beyond the negation of RESET_ for approximately 512 system clock cycles if the PLL is disabled. When the PLL is enabled, PORTC4 persists beyond the negation of RESET_ to allow for PLL lock for 100 microseconds times the ratio of the VCO to XTALA.



Reset timing parameters

Num	Description	Min	Typ	Max	Units
1	Power valid before reset negated	40			ms See note following table.
2	Reset asserted after power valid	3			T_{XTALA1}
3	Reset asserted while power valid	3			T_{XTALA1}
4	Reset asserted before power invalid	3			T_{XTALA1}

Note: RESET_ should remain low for at least 40ms after power reaches 3.0V.

SRAM timing

BCLK max frequency: 55.296 MHz

Operating conditions:

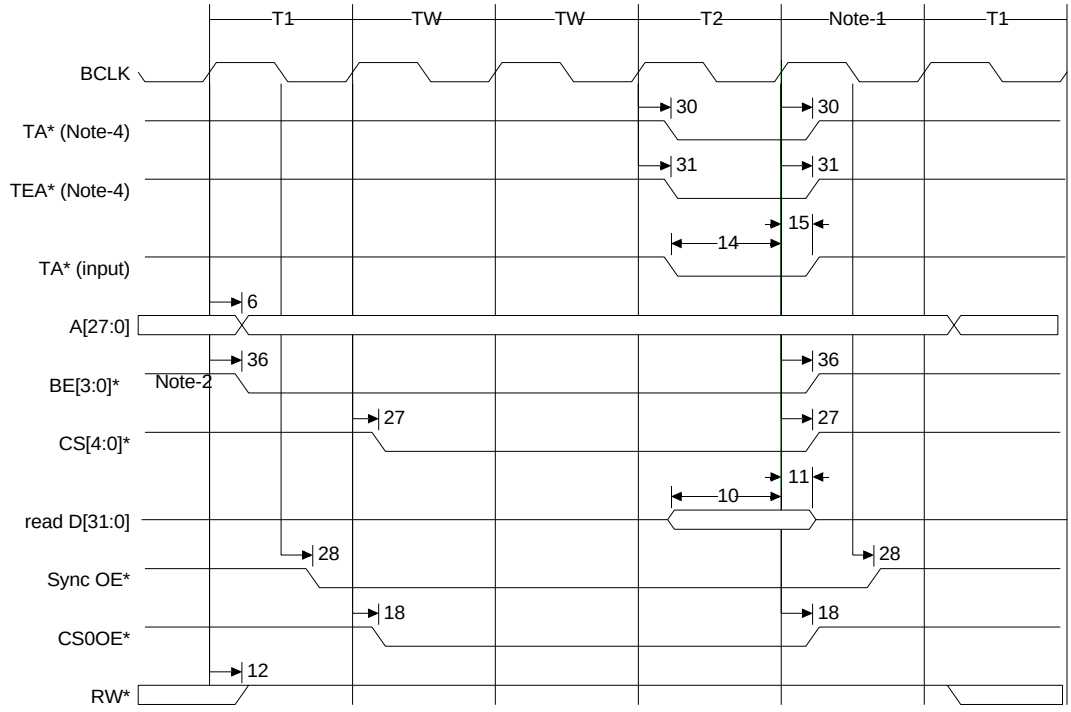
Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

SRAM timing parameters

Num	Description	Min	Max	Unit
36	BCLK high to BE* valid		15.5	ns
6	BCLK high to address valid	5	13.5	ns
9	BCLK high to data out valid		14	ns
13	BCLK high to data out high impedance		13	ns
10	Data in valid to BCLK high (setup)	5		ns
11	BCLK high to data in invalid (hold)	3		ns
14	TA* valid to BCLK high (setup)	5		ns
15	BCLK high to TA* invalid (hold)	3		ns
27	BCLK high to CS* valid		12.5	ns
28	BCLK low to OE* valid		12.5	ns
29	BCLK low to WE* valid		13	ns
30	BCLK high to TA* valid		13.5	ns
31	BCLK high to TEA* valid		16	ns
18	BCLK low to A27 (CS0OE*) valid		13.5	ns
19	BCLK low A26 (CS0WE*) valid		13.5	ns
12	BCLK high to RW* valid		13.5	ns

SRAM read

CS* controlled read (wait = 2)



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

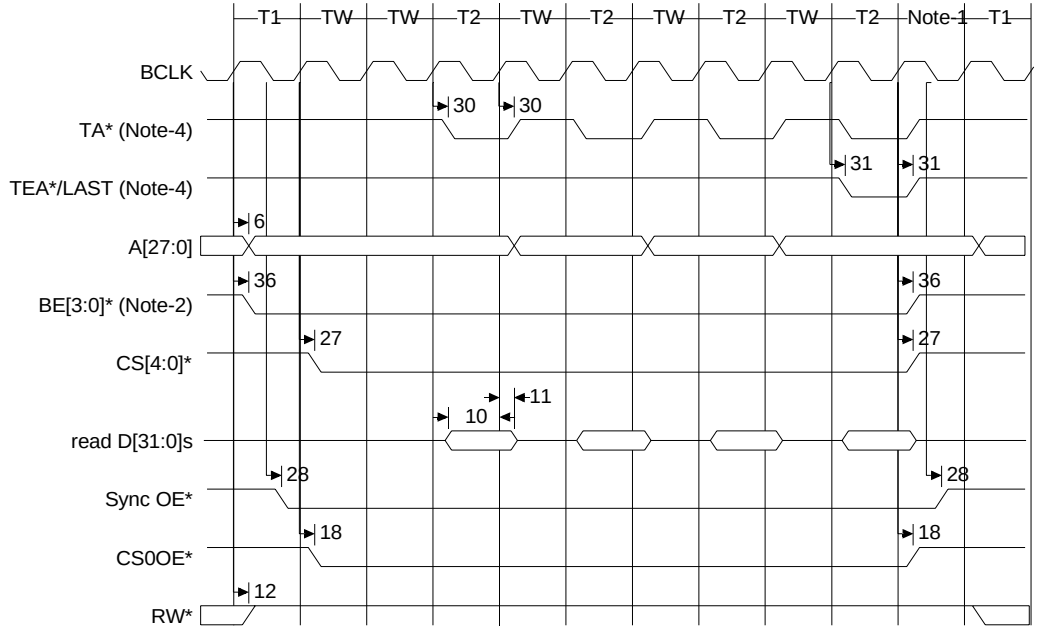
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM burst read

CS* controlled, four word (4-2-2-2), burst read (wait = 2, BCYC = 01)



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

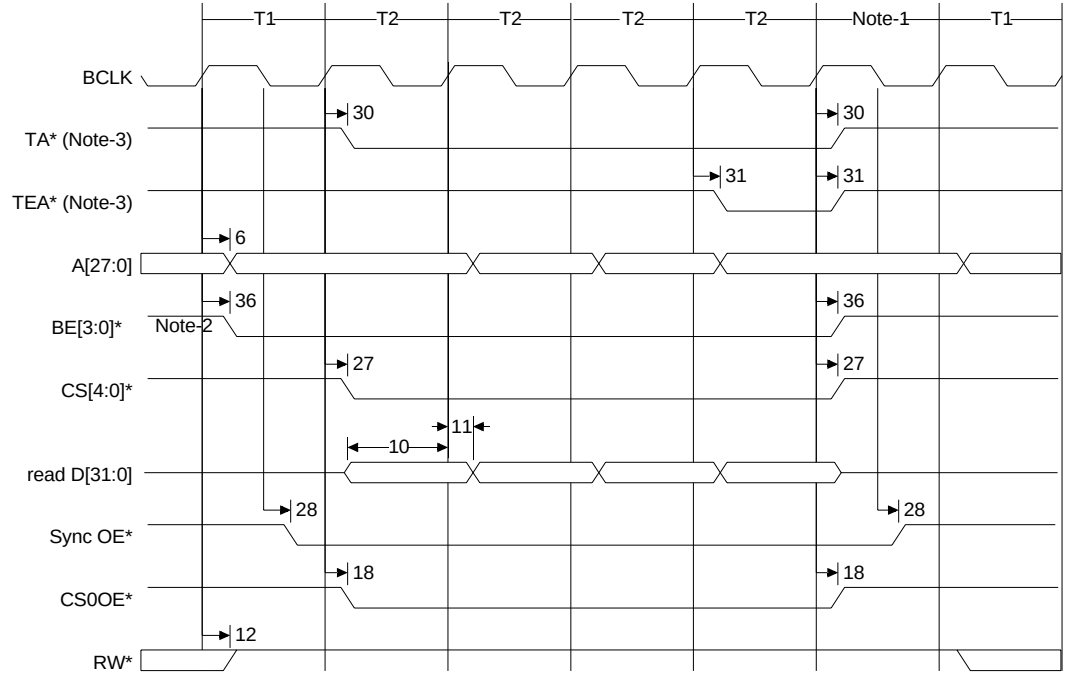
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM burst read (2111)

CS* controlled read (wait = 0, BCYC = 00)



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

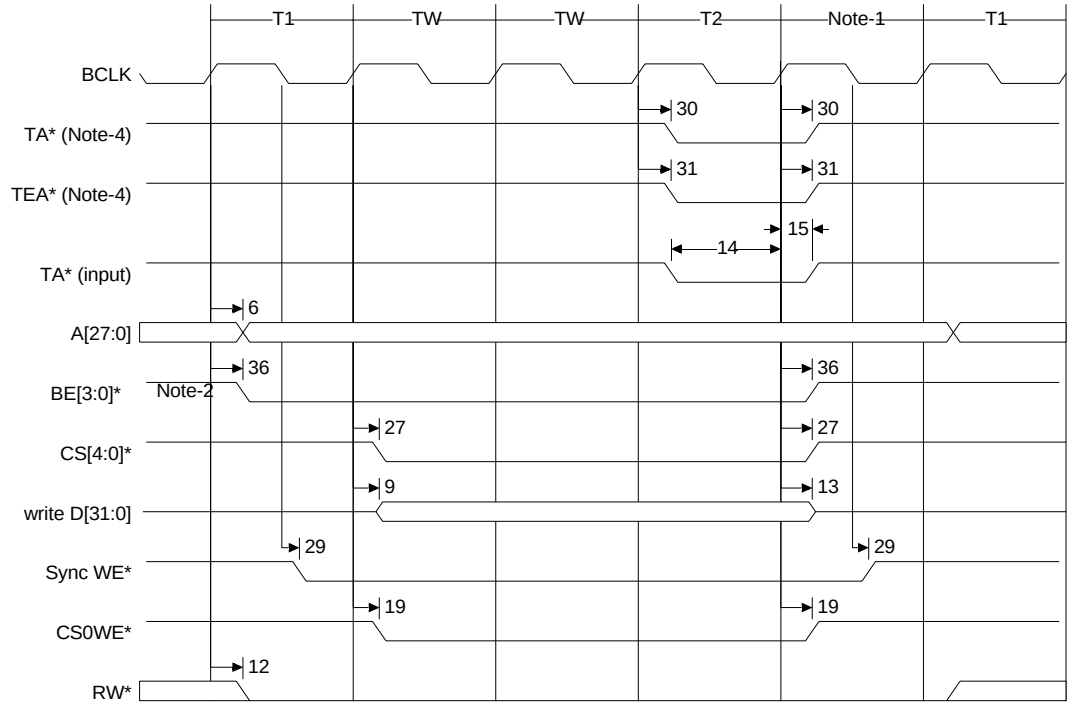
2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TA* and TEA*/LAST signals are for reference only.

SRAM write

CS controlled write (internal and external), (wait = 2)



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

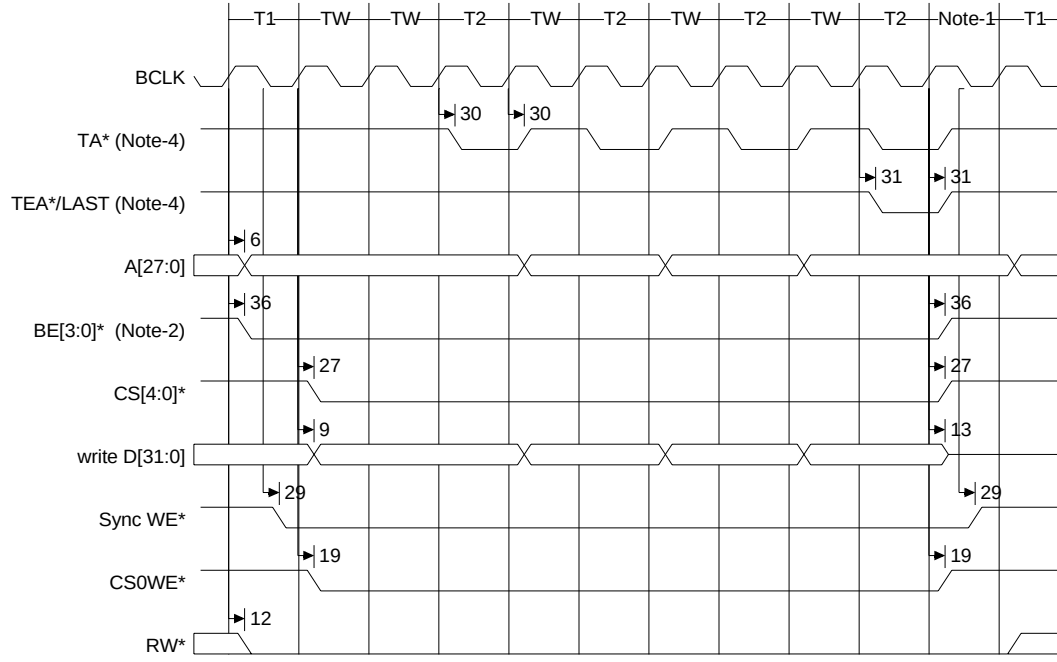
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM burst write

CS controlled, four word (4-2-2-2), burst write (wait = 2, BCYC = 01)



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

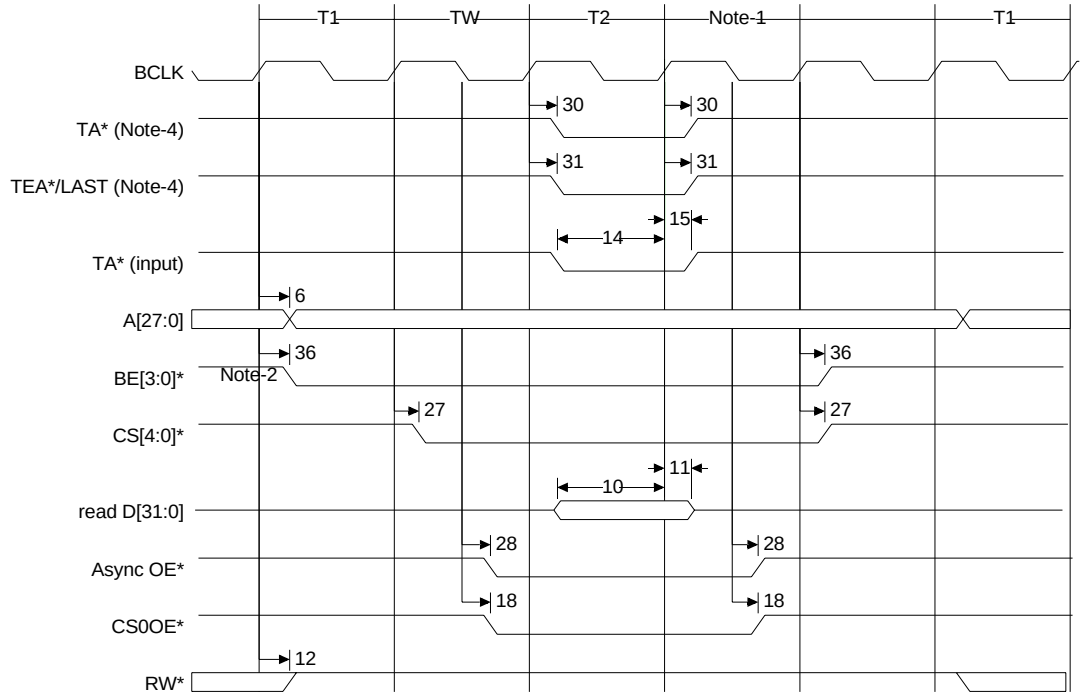
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM OE read

OE* controlled read (wait = 2)



Notes:

1 At least one null period occurs between memory transfers. More null periods can occur if the next transfer is DMA. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

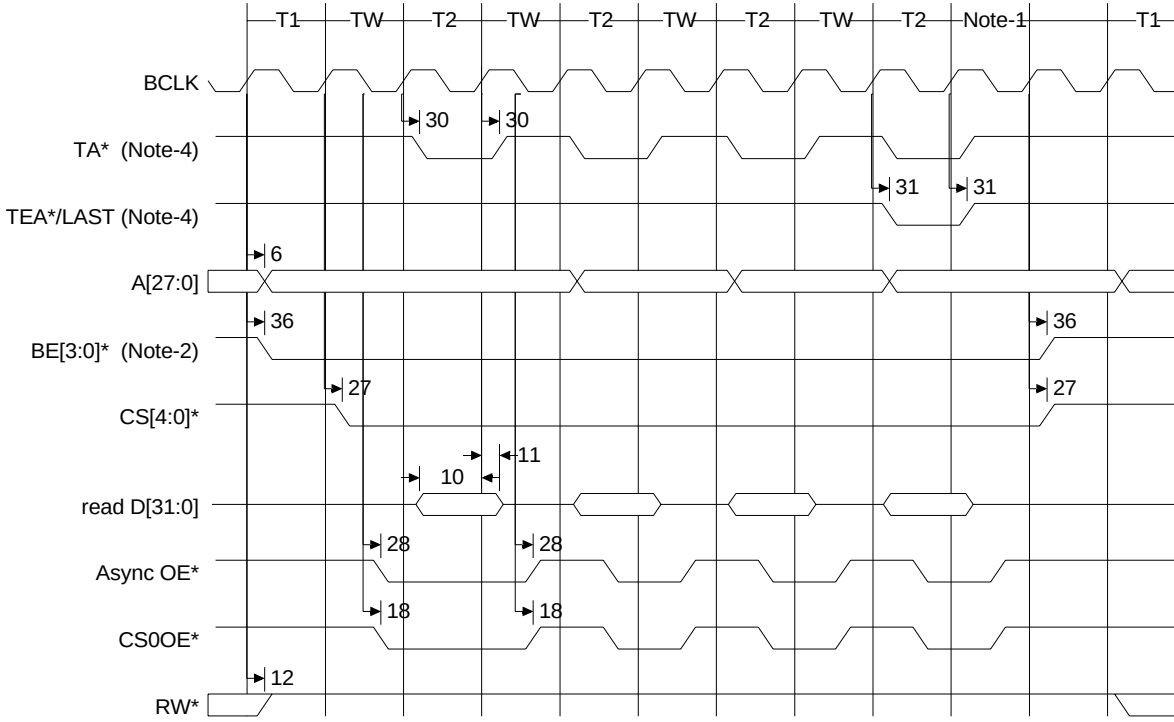
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM OE burst read

OE* controlled, four word (3-2-2-2), burst read (wait = 2, BCYC = 01)



Notes:

1 At least one null period occurs between memory transfers. More null periods can occur if the next transfer is DMA. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

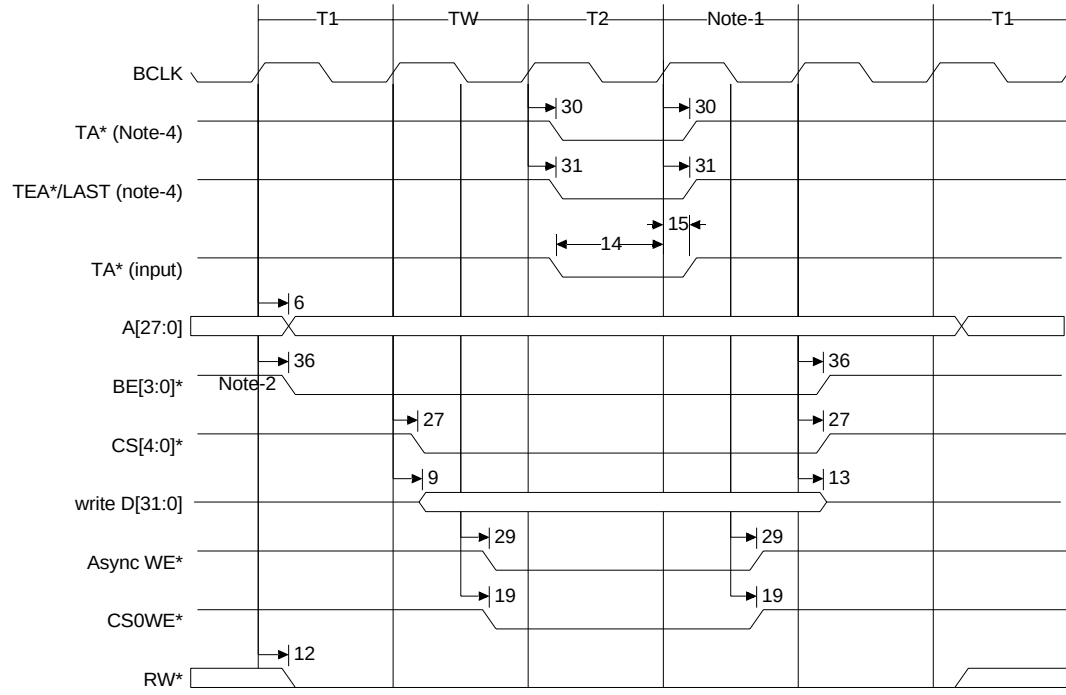
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM WE write

WE* controlled write (wait = 2)



Notes:

1 At least one null period occurs between memory transfers. More null periods can occur if the next transfer is DMA. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

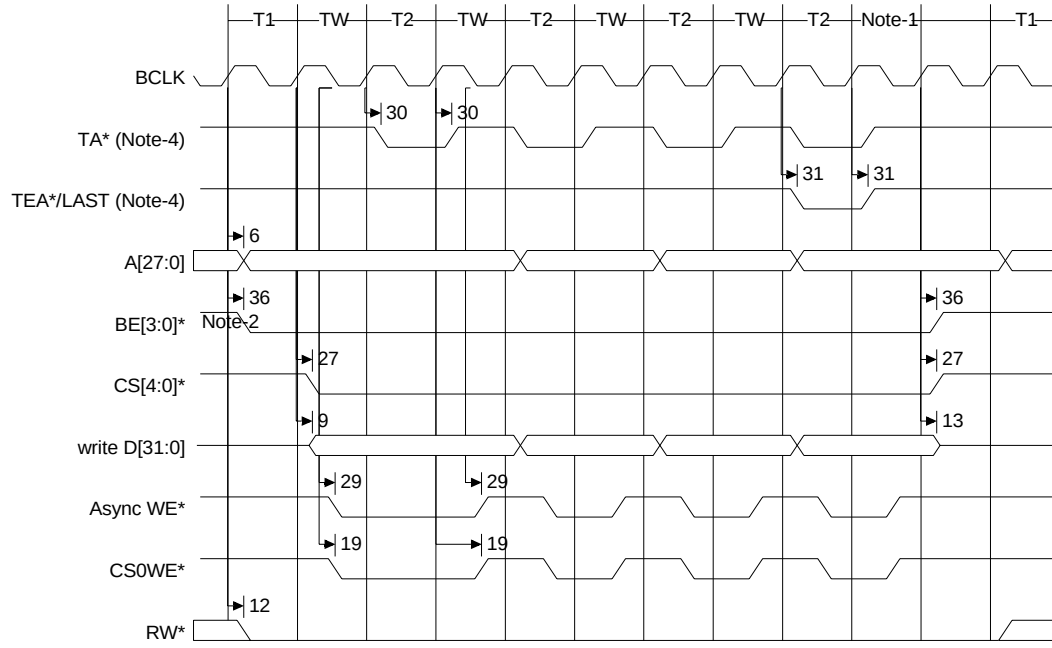
- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SRAM WE burst write

WE* controlled, four word (3-2-2-2), burst write (wait = 2, BCYC = 01)



Notes:

1 At least one null period occurs between memory transfers. More null periods can occur if the next transfer is DMA. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:0]
- 32-bit port = BE[3:0]

3 The TW cycles are present when the WAIT field is set to 2 or more.

4 The TA* and TEA*/LAST signals are for reference only.

SDRAM timing

BCLK max frequency: 55.296 MHz

Operating conditions:

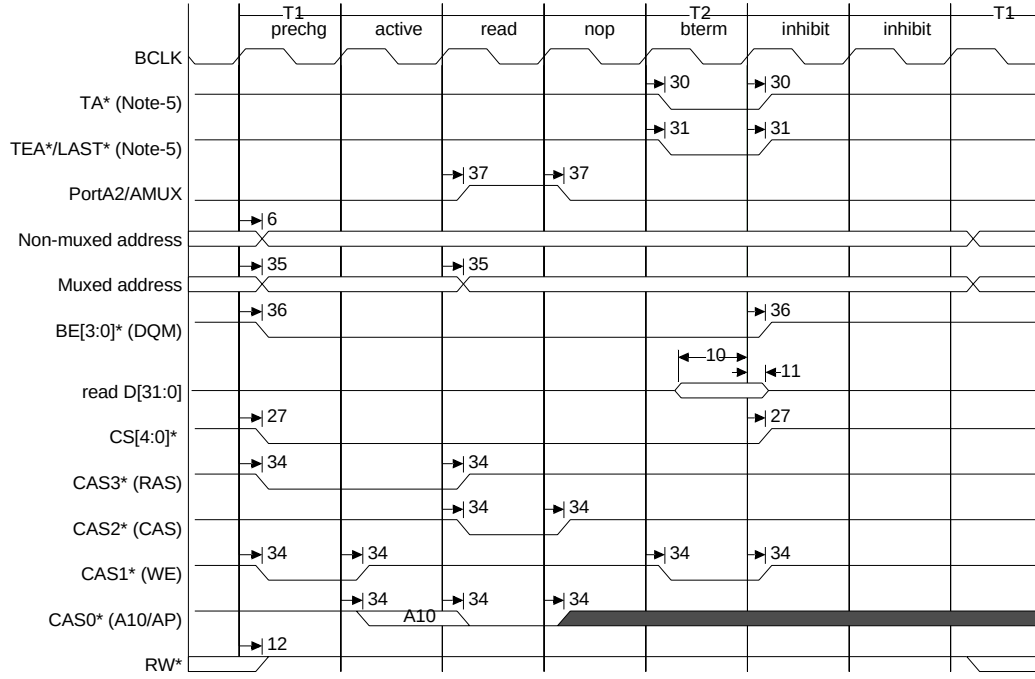
Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

SDRAM timing parameters

Num	Description	Min	Max	Unit
36	BCLK high to BE*/DQM* valid		15.5	ns
6	BCLK high to non-muxed address valid	5	13.5	ns
9	BCLK high to data out valid		14	ns
13	BCLK high to data out high impedance		13	ns
10	Data in valid to BCLK high (setup)	5		ns
11	BCLK high to data in invalid (hold)	3		ns
27	BCLK high to CS* valid		15.5	ns
30	BCLK high to TA* valid		13.5	ns
31	BCLK high to TEA* valid		16	ns
37	BCLK high to PORTA2/AMUX valid		14	ns
35	BCLK high to muxed address valid	6	14.5	ns
34	BCLK high to CAS* valid		12	ns
12	BCLK high to RW* valid		13.5	ns

SDRAM read

SDRAM read, CAS latency = 2



Notes:

1 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access.

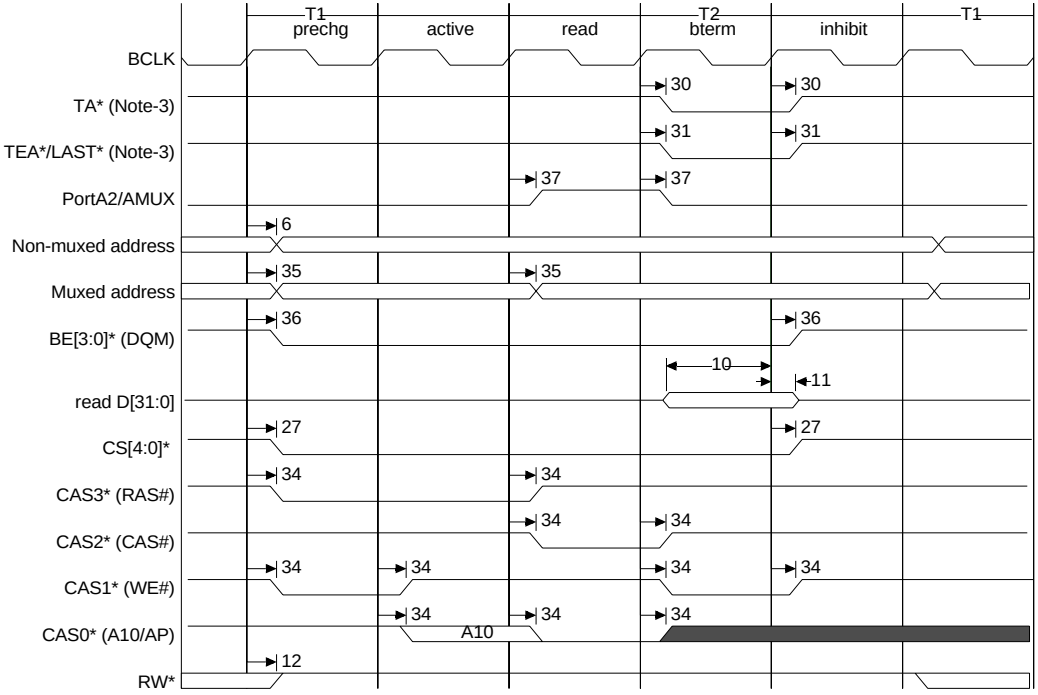
3 If CAS latency = 3, 2 NOPs occur between the read and burst terminate commands.

4 If CAS latency = 3, 3 inhibits occur after burst terminate.

5 The TA* and TEA*/LAST signals are for reference only.

SDRAM read

SDRAM read, CAS latency = 1



Notes:

1 Port size determines which byte enable signals are active:

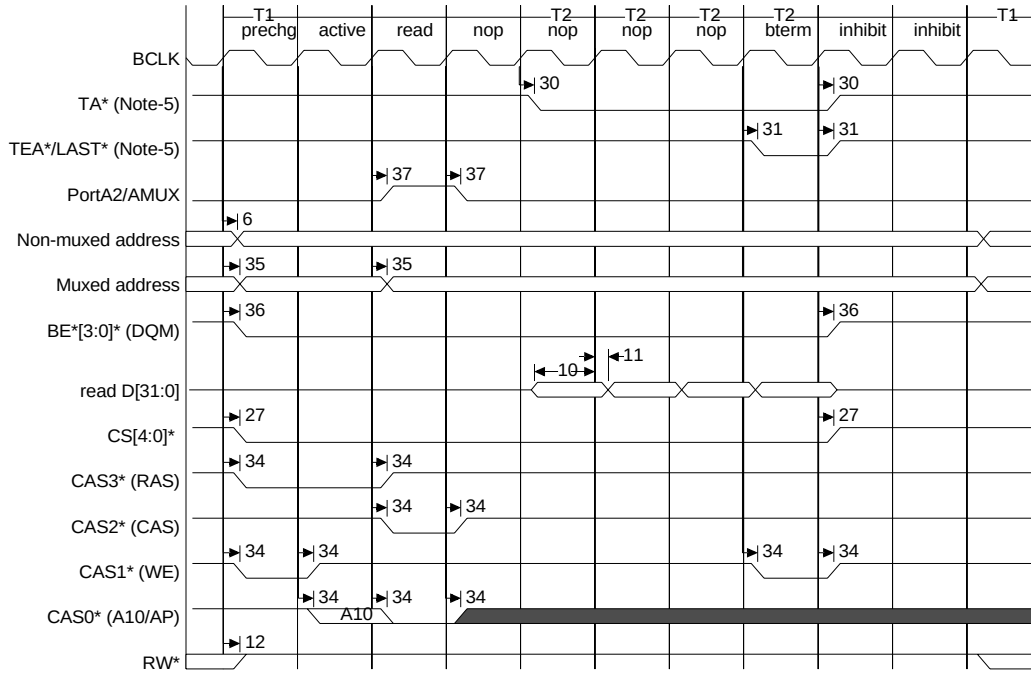
- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access.

3 The TA* and TEA*/LAST signals are for reference only.

SDRAM burst read

SDRAM read, CAS latency = 2



Notes:

1 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access.

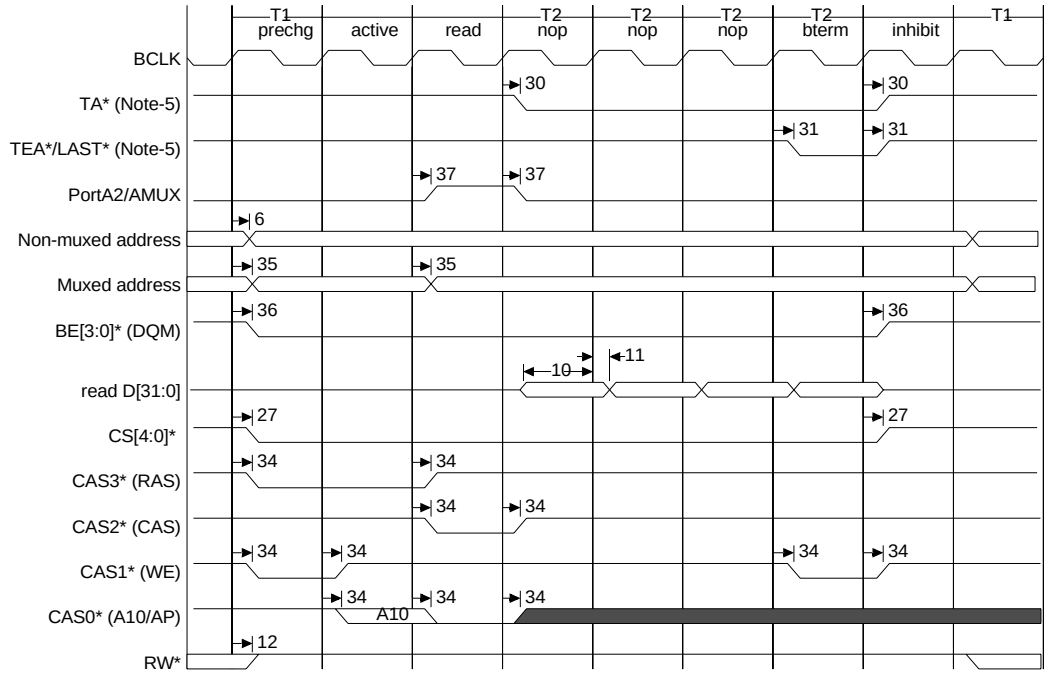
3 If CAS latency = 3, 5 NOPs occur between the read and burst terminate commands.

4 If CAS latency = 3, 3 inhibits occur after burst terminate.

5 The TA* and TEA*/LAST signals are for reference only.

SDRAM burst read

SDRAM read, CAS latency = 1



Notes:

1 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access.

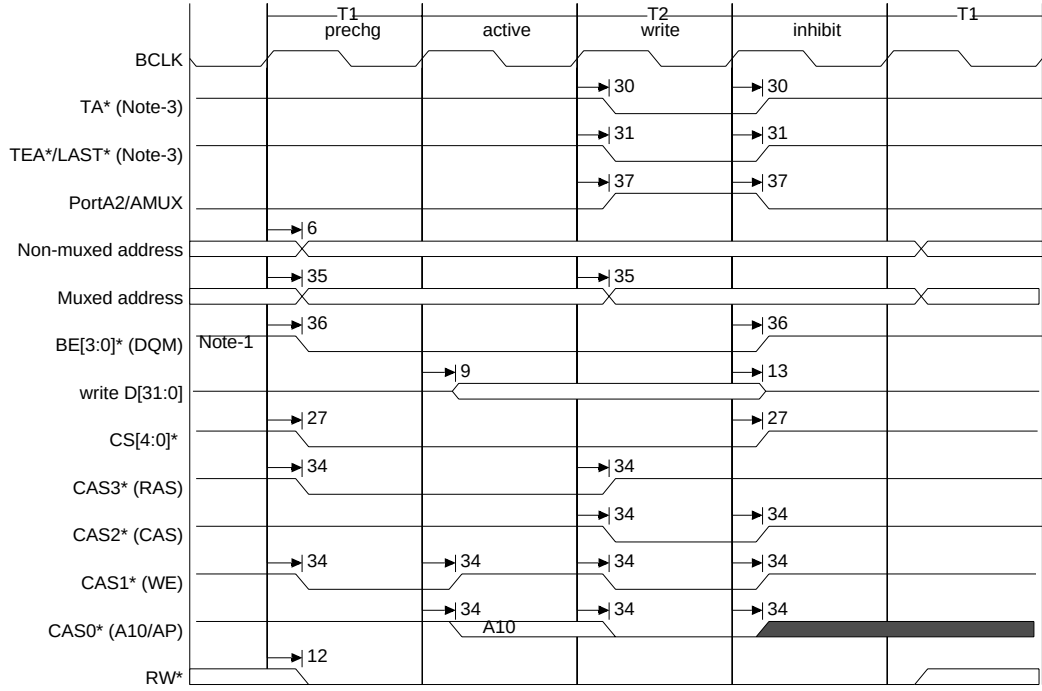
3 If CAS latency = 3, 5 NOPs occur between the read and burst terminate commands.

4 If CAS latency = 3, 3 inhibits occur after burst terminate.

5 The TA* and TEA*/LAST signals are for reference only.

SDRAM write

SDRAM write



Notes:

1 Port size determines which byte enable signals are active:

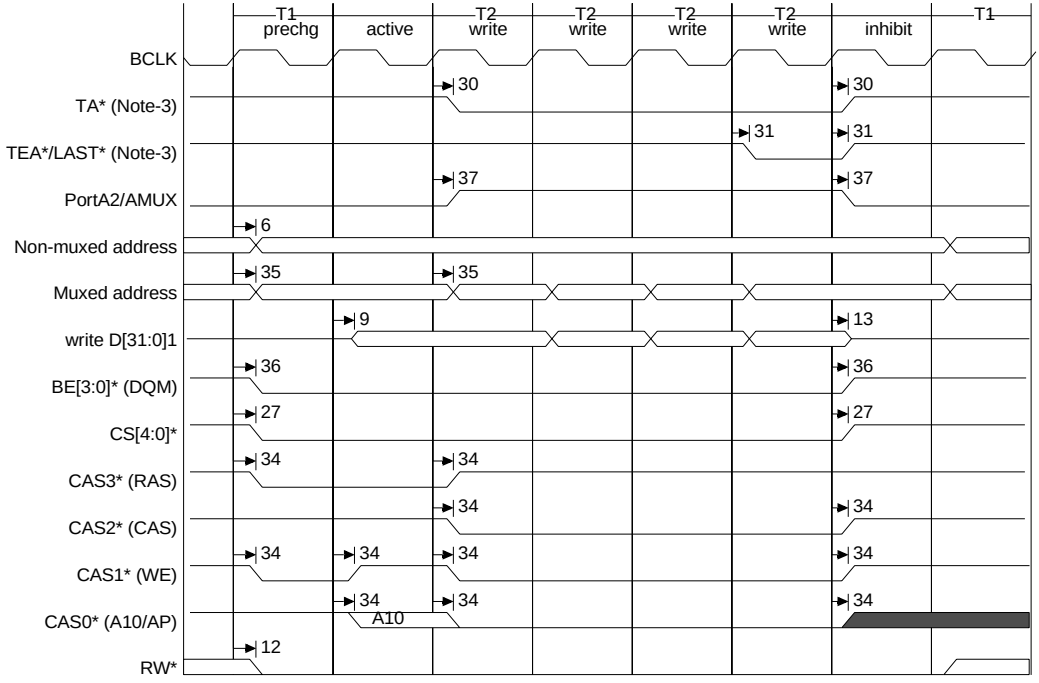
- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access.

3 The TA* and TEA*/LAST signals are for reference only.

SDRAM burst write

SDRAM burst write



Notes:

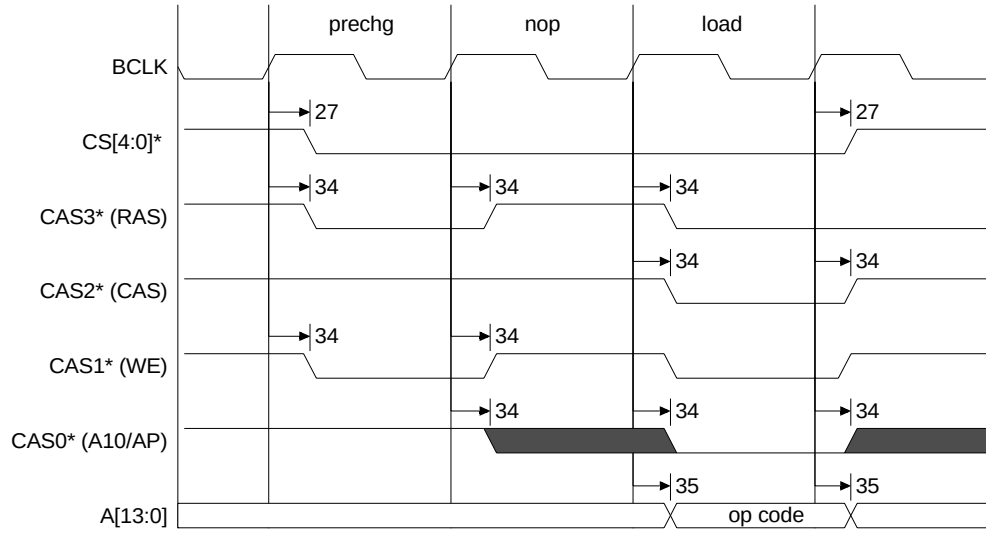
1 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

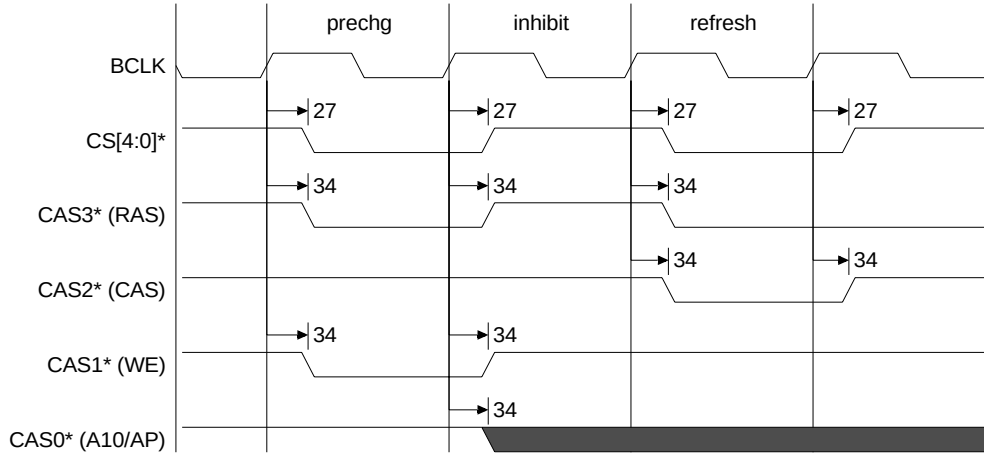
2 The precharge and/or active commands are not always present. These commands depend on the address of the previous SDRAM access. When the active command is not present, parameter #35 is valid during the write (T2) cycle.

3 The TA* and TEA*/LAST signals are for reference only.

SDRAM load mode



SDRAM refresh



FP DRAM timing

BCLK max frequency: 55.296 MHz

Operating conditions:

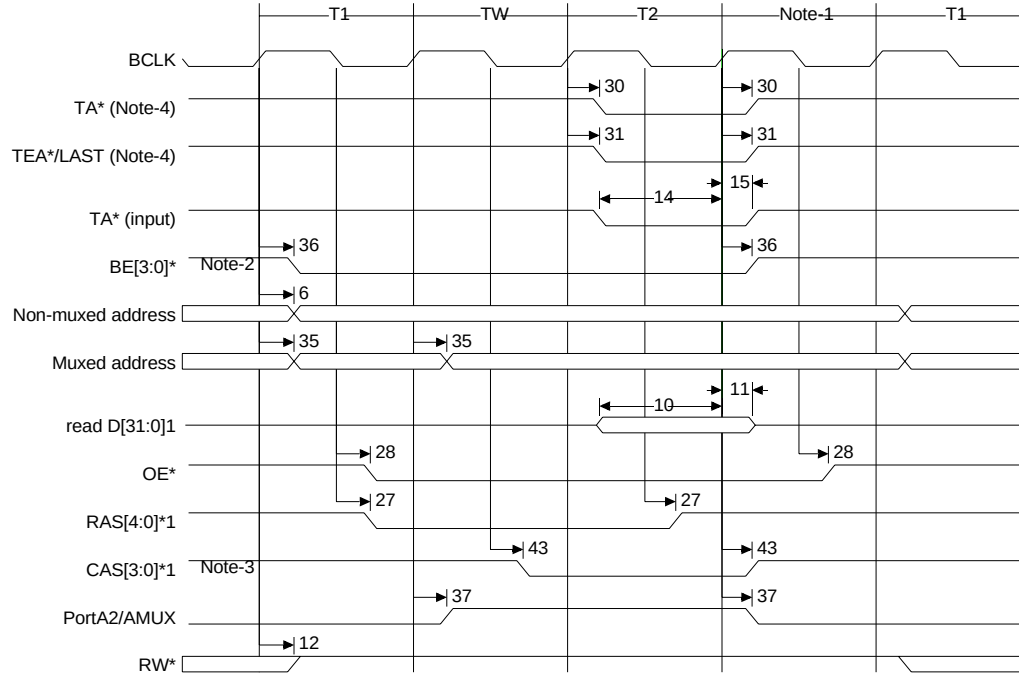
Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

FP DRAM timing parameters

Num	Description	Min	Max	Unit
36	BCLK high to BE* valid		15.5	ns
6	BCLK high to address valid	5	13.5	ns
9	BCLK high to data out valid		14	ns
13	BCLK high to data out high impedance		13	ns
10	Data in valid to BCLK high (setup)	5		ns
11	BCLK high to data in invalid (hold)	3		ns
14	TA* valid to BCLK high (setup)	5		ns
15	BCLK high to TA* invalid (hold)	3		ns
28	BCLK low to OE* valid		12.5	ns
29	BCLK low to WE* valid		13	ns
30	BCLK high to TA* valid		13.5	ns
31	BCLK high to TEA* valid		16	ns
37	BCLK high to PORTA2/AMUX valid		14	ns
35	BCLK high to muxed address valid	6	14.5	ns
43	BCLK low to CAS* valid		13	ns
27	BCLK low to RAS* valid		12	ns
12	BCLK high to RW* valid		13.5	ns

FP DRAM read

Fast Page read



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

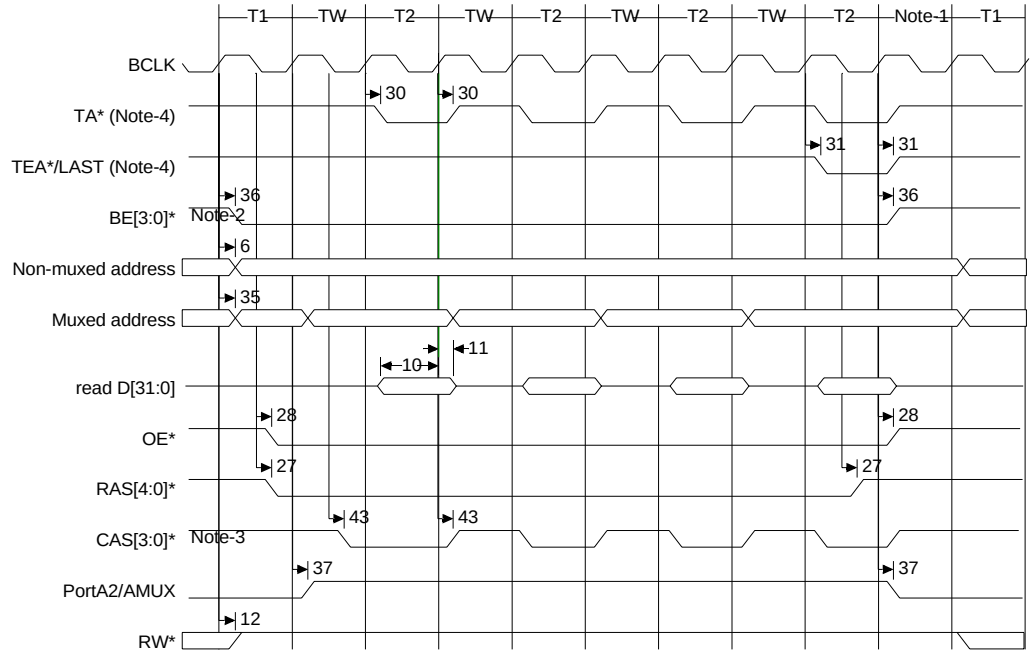
3 Port size determines which CAS signals are active:

- 8-bit port = CAS3*
- 16-bit port = CAS[3:2]
- 32-bit port = CAS[3:0]

4 The TA* and TEA*/LAST signals are for reference only.

FP DRAM burst read

Fast Page burst read



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

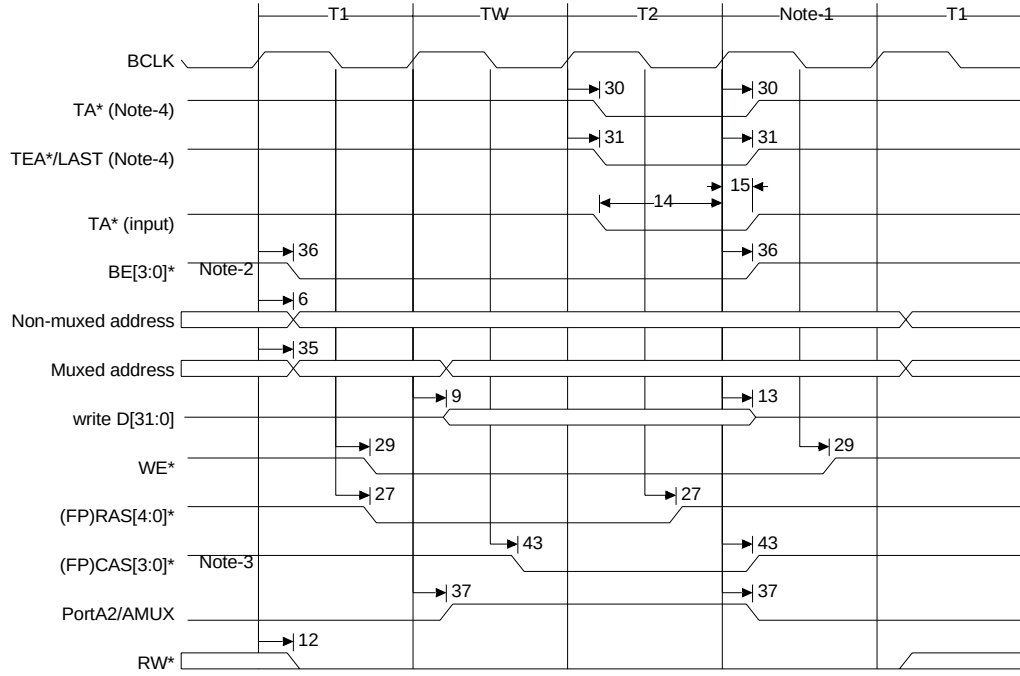
3 Port size determines which CAS signals are active:

- 8-bit port = CAS3*
- 16-bit port = CAS[3:2]
- 32-bit port = CAS[3:0]

4 The TA* and TEA*/LAST signals are for reference only.

FP DRAM write

Fast Page write



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

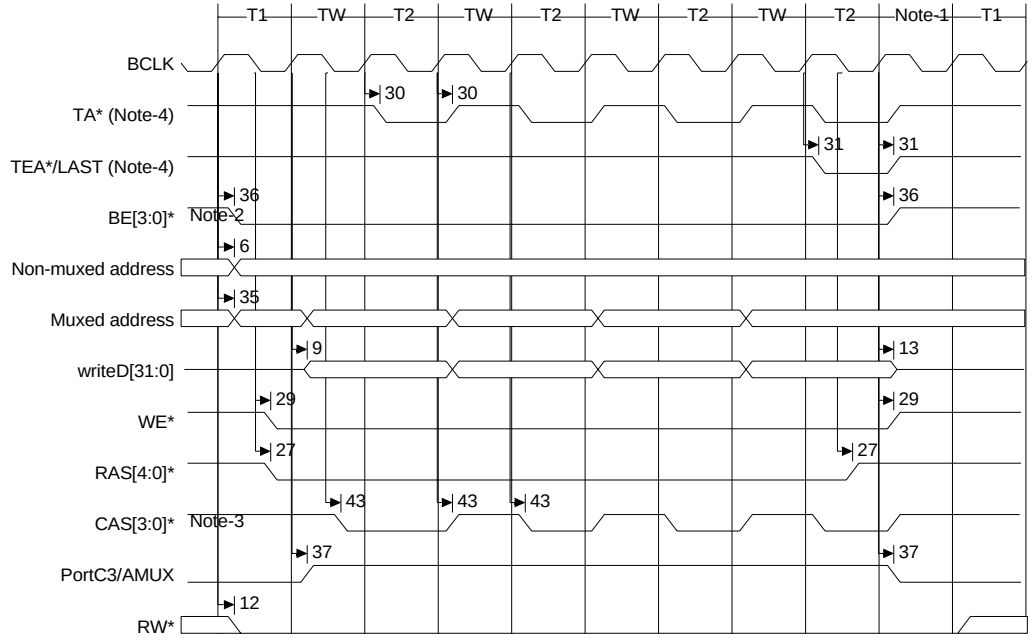
3 Port size determines which CAS signals are active:

- 8-bit port = CAS3*
- 16-bit port = CAS[3:2]
- 32-bit port = CAS[3:0]

4 The TA* and TEA*/LAST signals are for reference only.

FP DRAM burst write

Fast Page burst write



Notes:

1 If the next transfer is DMA, null periods between memory transfers can occur. Thirteen clock pulses are required for DMA context switching.

2 Port size determines which byte enable signals are active:

- 8-bit port = BE3*
- 16-bit port = BE[3:2]
- 32-bit port = BE[3:0]

3 Port size determines which CAS signals are active:

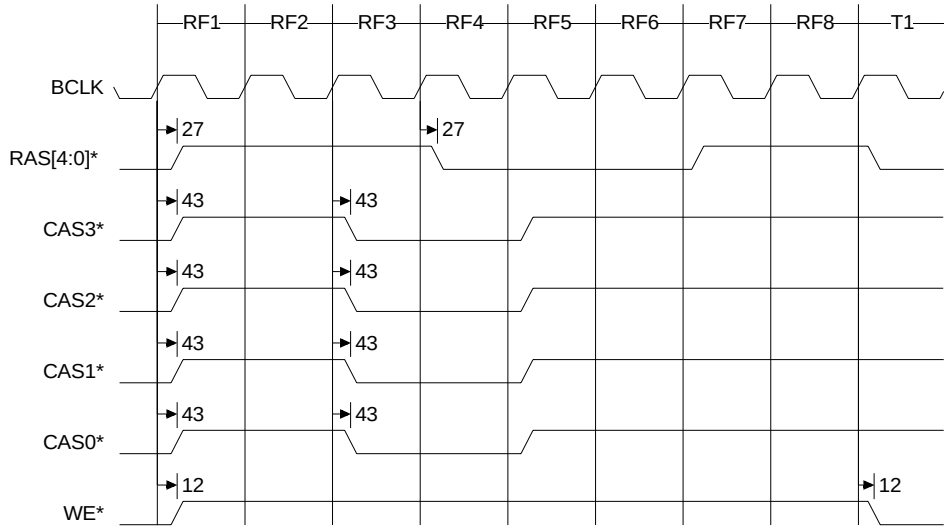
- 8-bit port = CAS3*
- 16-bit port = CAS[3:2]
- 32-bit port = CAS[3:0]

4 The TA* and TEA*/LAST signals are for reference only.

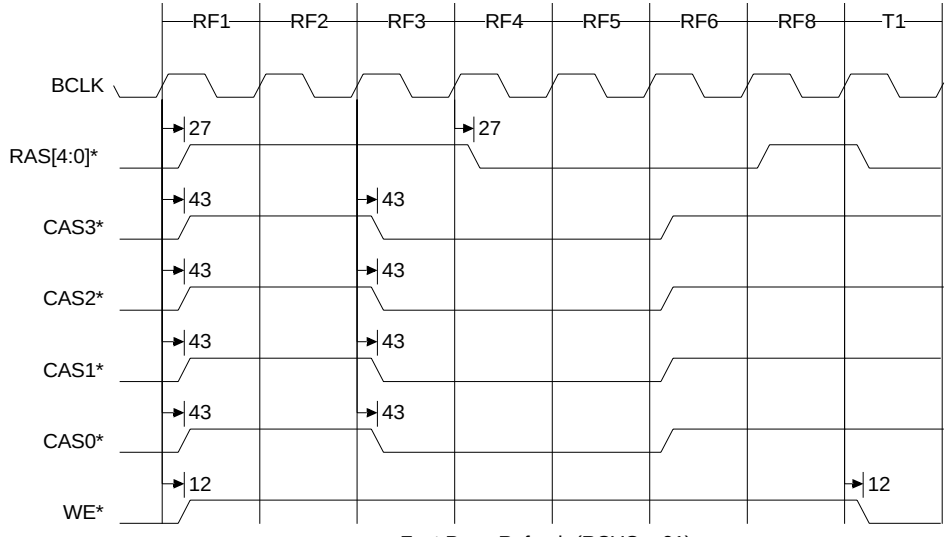
5 The BCYC field should never be set to 00.

fp_refresh_cycles

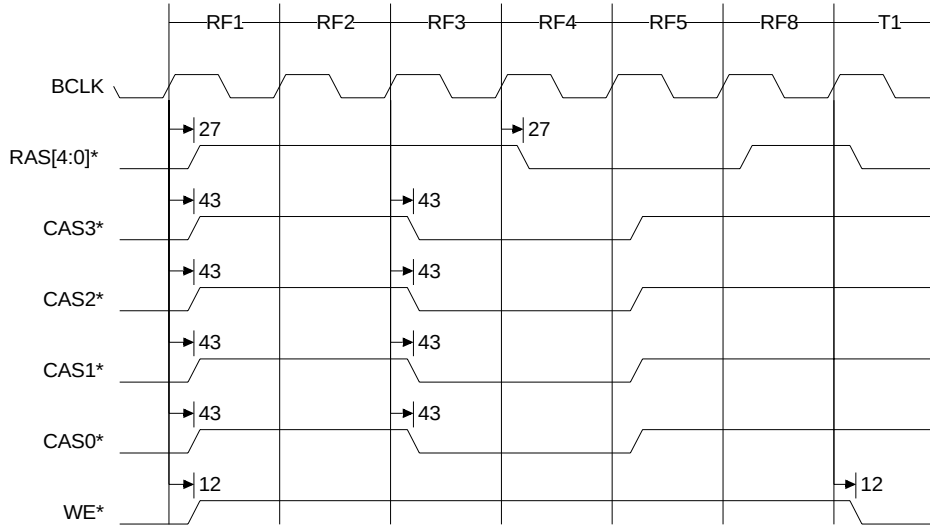
Fast page refresh (RCYC = 00)



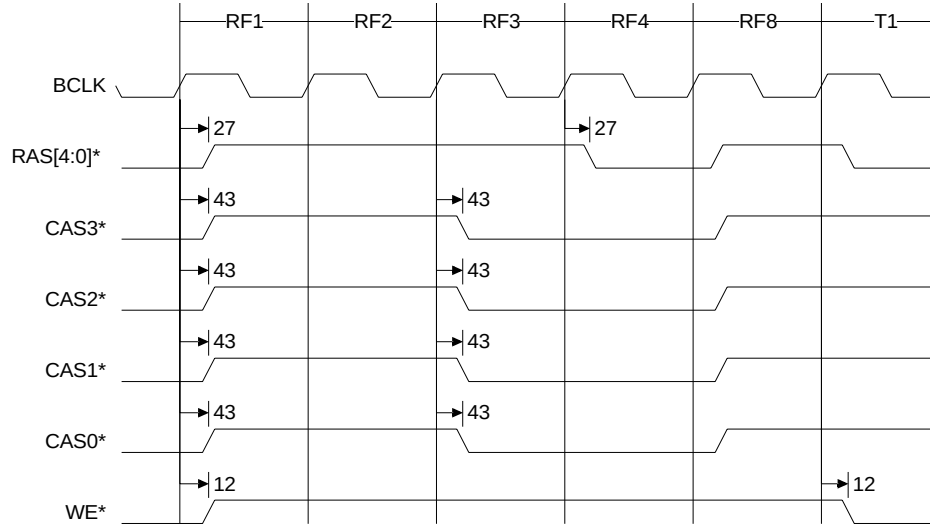
Fast page refresh (RCYC = 01)



Fast page refresh (RCYC = 10)



Fast page refresh (RCYC = 11)



Ethernet timing

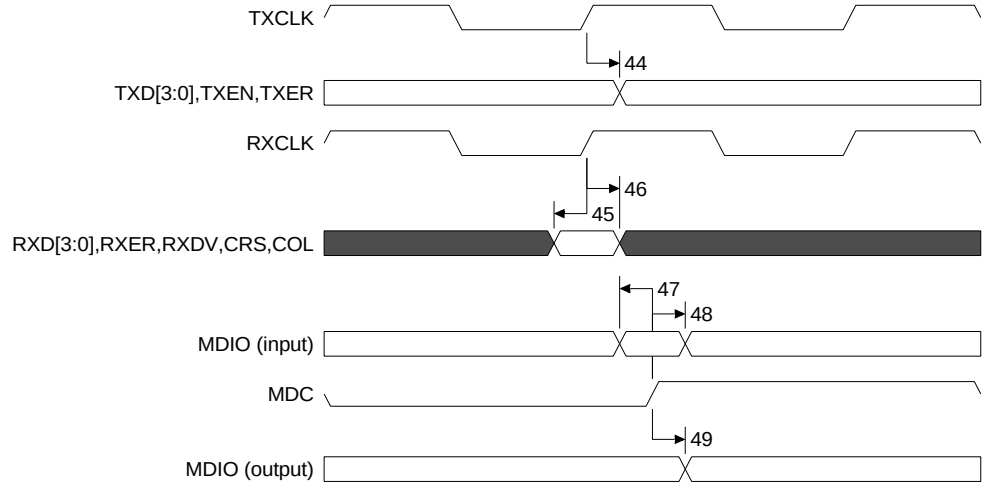
Operating conditions:

Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

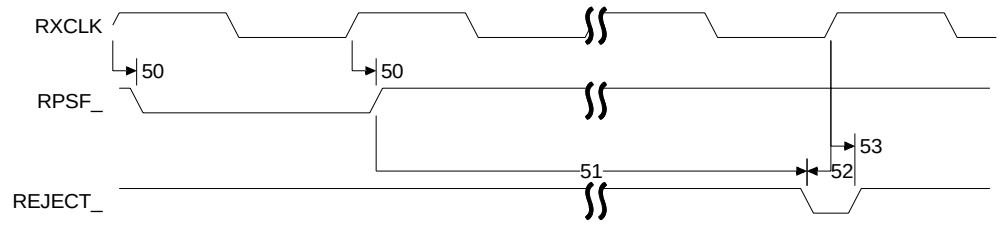
Ethernet timing parameters

Num	Description	Min	Max	Unit
44	TXCLK high to TXD*, TXEN, TXER valid		11.5	ns
45	RXD*, RXER, RXDV, TXCOL, RXCRS valid to RXCLK high (setup)	3		ns
46	RXCLK high to RXD*, RXER, RXDV, TXCOL, RXCRS hold time	2		ns
49	MDC high to MDIO valid		50	ns
47	MDIO valid to MDC high (setup)	3		ns
48	MDC high to MDIO hold time	3		ns
50	RXCLK high to RSPF* valid		15.5	ns
52	REJECT* valid to RXCLK high (setup)	3		ns
53	REJECT* valid from RXCLK high (hold)	1.5		ns

Ethernet PHY timing



Ethernet cam timing



JTAG timing

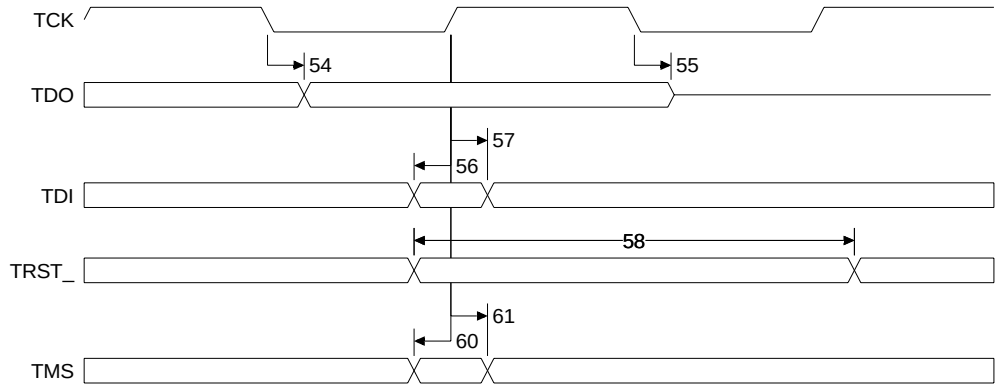
Operating conditions:

Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

jtag arm ice timing parameters

Num	Description	Min	Max	Units
54	TCK to TDO valid		21	ns
55	TCK to TDO HighZ		21	ns
56	TDI setup to TCK rising	1		ns
57	TDI hold from TCK rising	3		ns
58	TRST* width	1		T_{TCK}
60	TMS setup to TCK rising	1		ns
61	TMS hold from TCK rising	3		ns

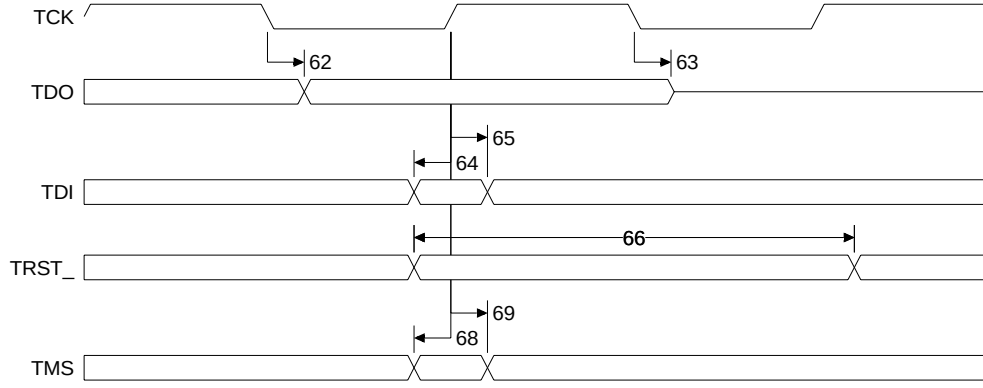
jtag arm ice timing diagram



jtag bscan timing parameters

Num	Description	Min	Max	Units
62	TCK to TDO valid		21	ns
63	TCK to TDO HighZ		21	ns
64	TDI setup to TCK rising	1		ns
65	TDI hold from TCK rising	3		ns
66	TRST* width	1		T_{TCK}
68	TMS setup to TCK rising	1		ns
69	TMS hold to TCK rising	3		ns

jtag bscan timing diagram



External DMA timing

BCLK max frequency: 55.296 MHz

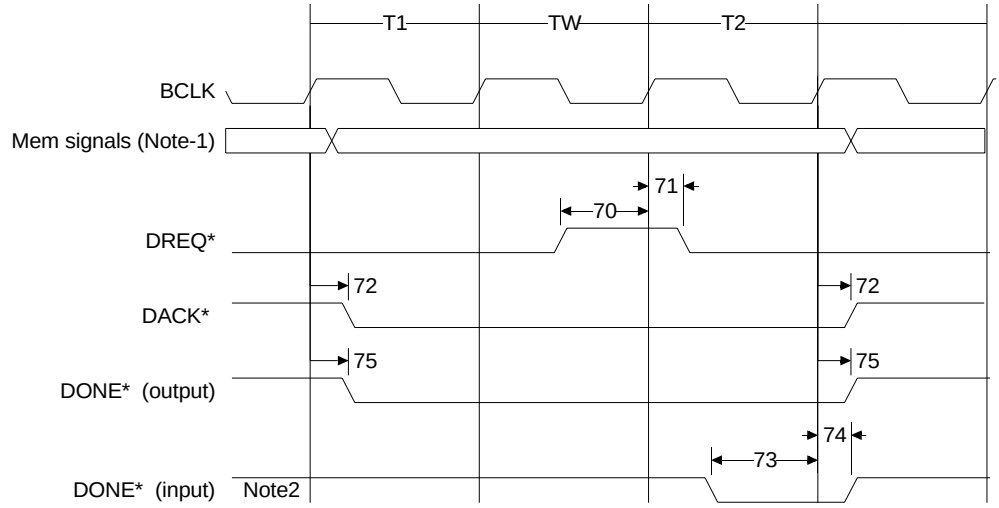
Operating conditions:

Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

External DMA timing parameters

Num	Description	Min	Max	Unit
72	BCLK high to DACK* valid		14	ns
75	BCLK high to DONE* (output) valid		15	ns
70	DREQ* low to BCLK high (setup)	5		ns
71	BCLK high to DREQ* valid (hold)	0		ns
73	DONE* (input) valid BCLK high (setup)	5		ns
74	BLCK high to DONE* (input) valid (hold)	0		ns

Fly-by external DMA

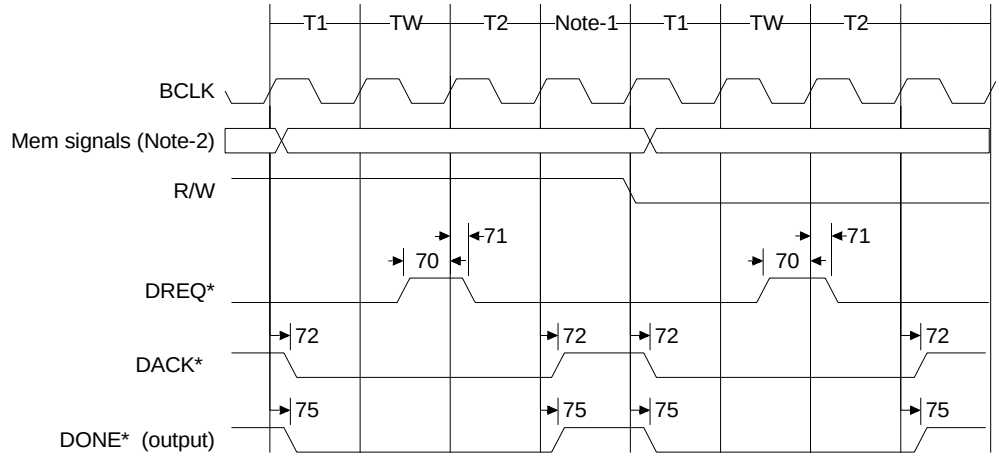


Notes:

1 The memory signals are data[31:0], addr[27:0], BE[3:0], CS/RAS[4:0], CAS[3:0], RW, OE*. WE*, and PORTC3/AMUX. The timing of these signals depends on how the memory is configured (Sync SRAM, Async SRAM, FP DRAM, or SDRAM).

2 The DONE* signal works as an input only when the DMA channel is configured as fly-by write.

Memory-to-memory external DMA



Notes:

1 A null period sometimes occurs between memory cycles.

2 The memory signals are data[31:0], addr[27:0], BE[3:0], CS/RAS[4:0], CAS[3:0], RW, OE*. WE*, and PORTA2/AMUX. The timing of these signals depends on how the memory is configured (Sync SRAM, Async SRAM, FP DRAM, or SDRAM).

Serial internal/external timing

Operating conditions:

Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

Note: SPI timing diagrams are in Chapter 10, "Serial Controller Module." See Figure 25, "SPI master mode 0 and 1 two-byte transfer," on page 219 and Figure 26, "SPI slave mode 0 and 1 two-byte transfer," on page 222. Only SPI modes 0 and 1 are supported.

Serial internal timing characteristics

Num	Description	Min	Max	Unit
76	SCLK to ENABLE high	1		T_{SCLK}
77	SCLK to TXD (PORTA7/C7)		1 T_{SYS}*	ns
78	RXD (PORTA3/C3) setup to SCLK	1		ns
79	RXD hold to SCLK	1		ns

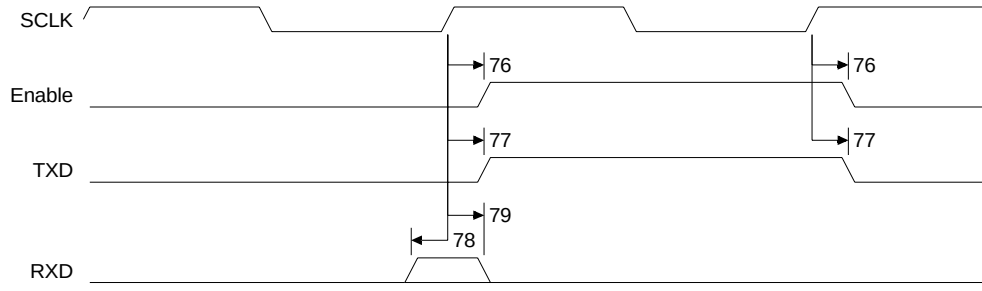
* The T_{SYS} parameter represents one period of the internal system clock.

Serial external timing characteristics

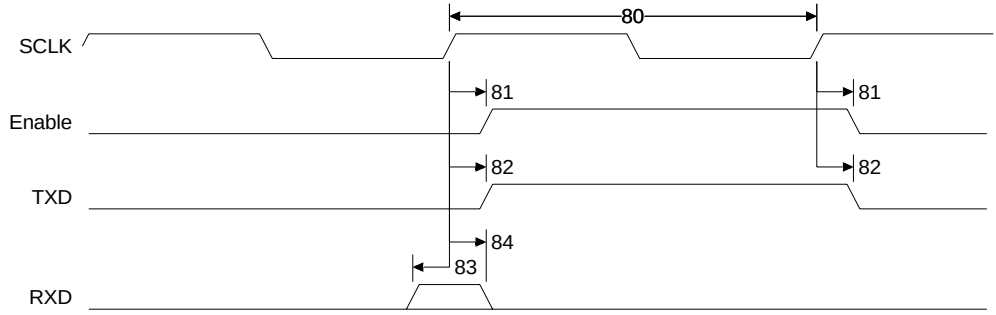
Num	Description	Min	Max	Unit
80	SCLK frequency		10	MHz
	SCLK duty cycle	45	55	%
81	SCLK to ENABLE	1		T_{SCLK}
82	SCLK to TXD (PORTA7/C7)		$2T_{SYS}$*	ns
83	RXD (PORTA3/C3) setup to SCLK	2		ns
84	RXD hold to SCLK	1.5		ns

* The T_{SYS} parameter represents one period of the internal system clock.

synchronous serial internal clock



synchronous serial external clock



GPIO timing

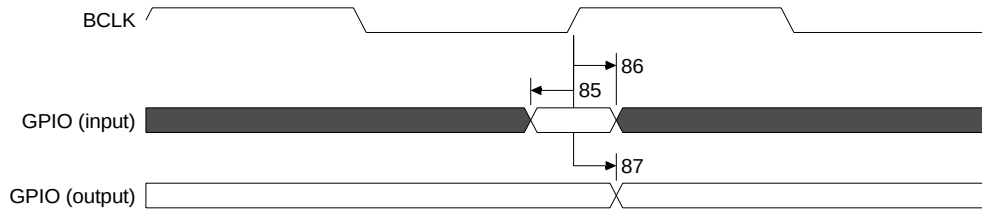
Operating conditions:

Temperature:	-15.00 (min)	110.00 (max)
Voltage:	1.60 (min)	1.40 (max)
Output load:	25.0pf	
Input drive:	CMOS buffer	

GPIO timing parameters

Num	Description	Min	Max	Unit
85	GPIO (setup) to BCLK rising	3		ns
86	GPIO (hold) from BCLK rising	0		ns
87	BCLK to GPIO (output)		17	ns

GPIO timing diagram



Index

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W


Digi®